



Département Mécanique
Cycle d'Ingénieur : Mécatronique-Productique
Semestre S3
Modules : Mécatronique 1 et 2

Conception, modélisation et contrôle d'un Pendule inverse



Réalisé par :

IDRISSI hamza

BANANE Mohamed

CHAMIT Salim

TACHTIOUINE Hicham

BELAISSAOUI Larbi

Encadrant : Mr. Zakaria KHAOUCH

Année universitaire 2023/20

Table des matières

Introduction Générale

Chapitre I : Généralité sur les pendules inversés

I.	Introduction	5
II.	Pendule inversé.....	7
1.	Pendule inversé simple effet ou pendule simple classique	7
2.	Pendule inversé double effet.....	7
3.	Pendule inversé rotatif	8
III.	Quelques applications du pendule inversé.....	8
IV.	Conclusion.....	9

Chapitre II : Conception , Modélisation et simulation du système

I .	Description du système.....	11
1.	Principe de fonctionnement	11
II .	Conception de système avec SolidWorks	12
II .	Modélisation du système (chariot-pendule)	13
1.	Modélisation et régulation d'un pendule inverse	14
a)	Conception du contrôleur	14
b)	Modèle mathématique de pendule inverse.....	15
3.	Modélisation par Matlab Simulink(Schéma Block).....	19
4.	Modélisation par 20 – Sim(Schéma block).....	31
5.	Modélisation avec langage python.....	39

Chapitre III : Réalisation et test

I.	Matériel utilisé	43
1.	Support	43
2.	Actionneur	43
3.	Microcontrôleur	43
4.	Capteur	44
5.	Chariot	44
II.	Environnement de développement	45
1.	IDE Arduino	45
2.	LabVIEW.....	46
	✓ Introduction au Bloc Diagramme LabVIEW pour le Pendule Inverse.....	46
	✓ Stabilisation du Système.....	47
	✓ Introduction au Front Panel LabVIEW pour le Pendule Inverse	48
	✓ Interface Homme-Machine (IHM)	49

Chapitre VI. Conclusion Générale

Table des figures

Figure 1: synoptique de banc d'essais du pendule inverse.....	5
Figure 2: Pendule inversé simple effet	7
Figure 3: Double pendule inverse.....	7
Figure 4: Pendule inverse rotatif	8
Figure 5: HONDA RIDING ASSIST-E MOTORCYCLE	8
Figure 6: HONDA RIDING ASSIST-E MOTORCYCLE	9
Figure 7: Segway	9
Figure 8: Hoverboard.....	9
Figure 9: châssis	11
Figure 10: Pendule.....	12
Figure 11: Car	12
Figure 12: Encodeur	13
Figure 13: Motor DC avec poulie.....	13
Figure 14: pendule inverse.....	14
Figure 15: montre le schéma d'un contrôleur PID	15
Figure 16: Schéma du contrôleur LQR.....	15
Figure 17: pendule inverse dans SIMSCAPE	19
Figure 18: le bloc du chariot	20
Figure 19: le bloc du pendule	20
Figure 20: le bloc des joints	21
Figure 21: le bloc des capteurs	21
Figure 22: le bloc des actionneurs	22
Figure 23: le bloc de l'environnement	22
Figure 24 : le bloc des bases	22
Figure 25: pendule inverse sans PID	22
Figure 26 : le résultat	23
Figure 27: pendule inverse avec PID	24
Figure 28 : l'opération de "tuning"	24
Figure 29: le résultat	25
Figure 30: la fonction de Transfer	26
Figure 31: un cadre mondial	26
Figure 32: le bloc du joint de translation	27
Figure 33: le bloc du chariot	27
Figure 34: le chariot dans de la pendule dans simscape.....	28
Figure 35: l'articulation pivot	28
Figure 36 : le bloc de pendule	28
Figure 37: le schéma physique de pendule inverse dans Simscape	29
Figure 38: le résultat du simulation sans PID.....	29
Figure 39: pendule inverse avec le contrôleur PID	29
Figure 40: le résultat du simulation avec PID	30
Figure 41: Schéma block du système	35
Figure 42: PID_Sous_Block	35
Figure 43: PID_Maximal_Value_Sous_Block.....	36
Figure 44: Variation de l'angle Théta / Déplacement du chariot	36
Figure 45: Forme finale du système dans l'éditeur 3D	37
Figure 46: Génération du Code 20-Sim	37
Figure 47: Model Final	38
Figure 48: Simulation réelle par Python	41
Figure 49: Résultat obtenu	41
Figure 50: Simulation réelle par Python	41
Figure 51: Structure	43
Figure 52: Moteur DC	43
Figure 53: Carte Arduino uno	44

Figure 54: Encodeur rotatif	44
Figure 55 : Chariot.....	44
Figure 56: IDE Arduino	45
Figure 57: Le bloc Diagramme du code LabVIEW	46
Figure 58: Face avant LabVIEW	49
Figure 59: Istration de la Communication Arduino- LabVIEW	50

Introduction Générale

Le pendule inversé, également connu sous le nom de pendule inversé à contrôle actif, représente un domaine fascinant et crucial dans le domaine de la dynamique des systèmes et du contrôle automatique. Ce système mécanique complexe se distingue par sa capacité à maintenir un équilibre instable en utilisant des techniques de contrôle sophistiquées. Le pendule inversé est une illustration emblématique des principes fondamentaux du contrôle dynamique, offrant des défis stimulants pour les ingénieurs et les chercheurs travaillant dans le domaine de la robotique et du contrôle automatique.

Dans cette introduction, nous explorerons les caractéristiques fondamentales du pendule inversé, son importance dans divers domaines d'application, ainsi que les avancées récentes dans le développement de stratégies de contrôle avancées. En comprenant les complexités inhérentes à ce système, nous serons en mesure d'apprécier les efforts déployés pour améliorer sa stabilité et son comportement dynamique, ouvrant ainsi la voie à des applications pratiques dans des domaines tels que la robotique industrielle, la stabilisation des véhicules et les technologies d'assistance humaine.

Joignez-vous à nous pour un voyage captivant à travers les mécanismes du pendule inversé, où la fusion de la théorie du contrôle et de la mécanique avancée crée des opportunités passionnantes pour l'innovation technologique.

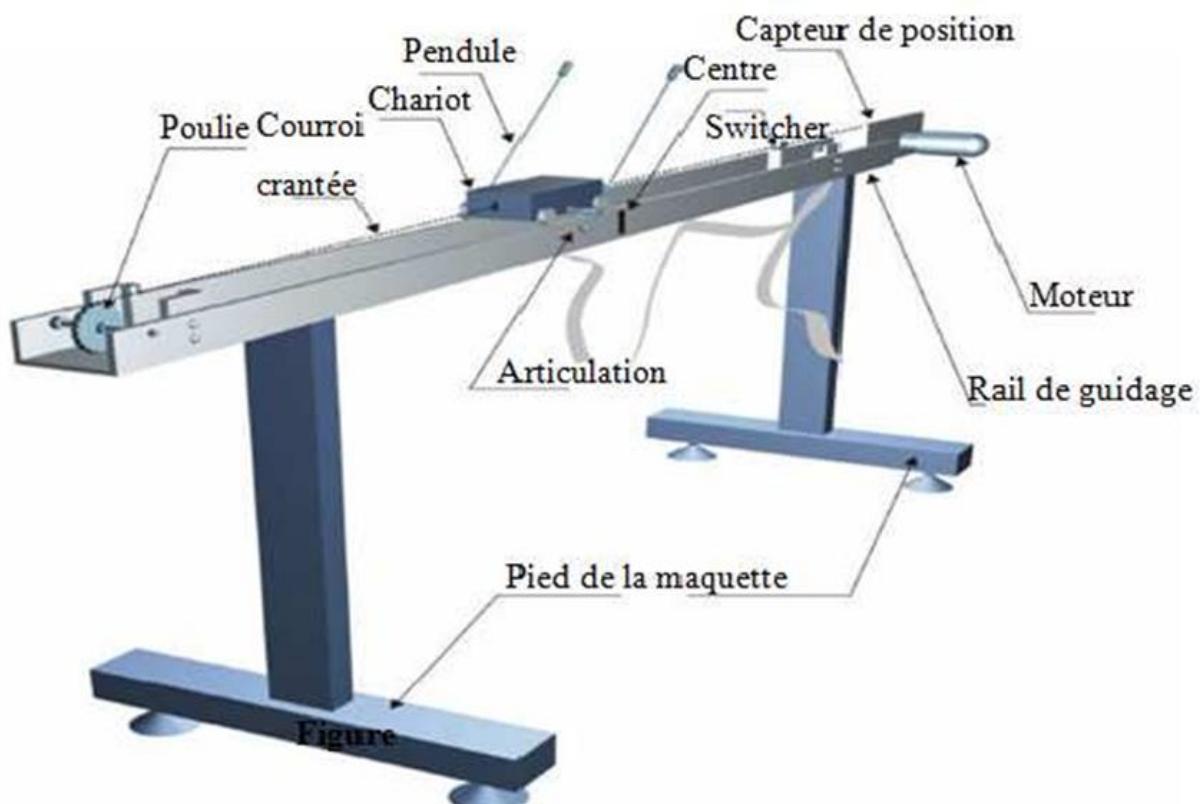


Figure 1: synoptique de banc d'essais du pendule inverse

CHAPITRE I

Généralité sur les pendules inverses

I. Introduction

Dans ce chapitre, nous présentons dans un premier temps quelques types des pendules notamment les pendules inversés. Nous nous intéressons par la suite à la représentation de quelques exemples de systèmes réels inspirés et basés sur le principe des pendules inversés

II. Pendule inverse

Souvenez-vous quand vous étiez petit et que vous essayiez de tenir en équilibre une tige survotre doigt. Vous deviez en permanence ajuster la position de votre main pour stabiliser la tige. Le pendule inversé fait la même chose.

Il existe plusieurs types de pendules inversés, chacun avec ses propres caractéristiques et applications. Voici quelques exemples :

1. Pendule inverse simple effet ou pendule simple classique

Ce type de pendule inversé est le plus simple. Il se compose d'une tige rigide fixée par un point de pivot en bas et une masse suspendue au-dessus du point de pivot. Lorsque la masse est déplacée, le pendule inversé utilise un mécanisme de contrôle pour maintenir son équilibre en ajustant sa position.

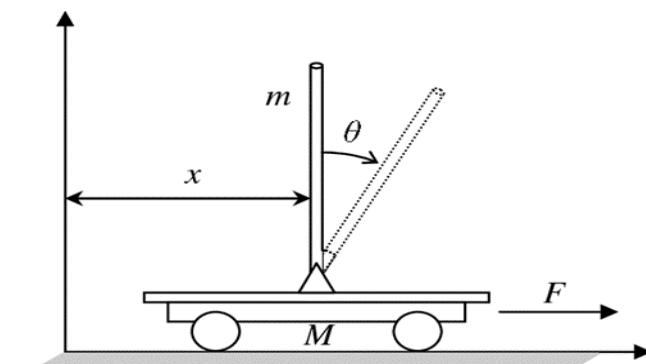


Figure 2: Pendule inversé simple effet

2. Pendule inverse double effet

Le pendule inversé à double effet est un système plus complexe. Il comporte deux masses suspendues à deux points de pivot distincts, l'une au-dessus et l'autre en dessous du point de pivot central. Ce type de pendule inversé peut être utilisé pour des applications plus avancées nécessitant un contrôle plus précis.

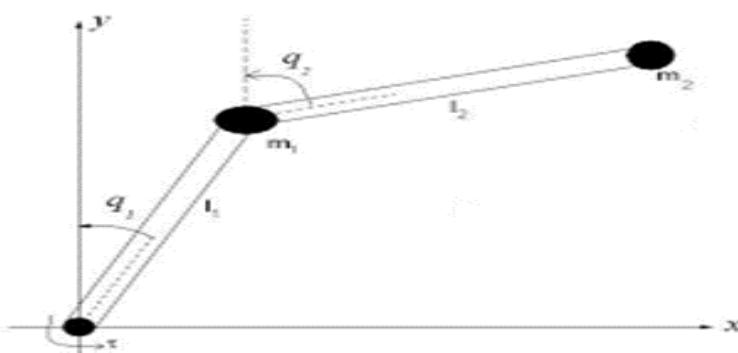


Figure 3: Double pendule inverse

3. Pendule inverse rotatif

Ce pendule est composé d'un bras actionné en rotation dans le plan horizontal, à son extrémité vient s'ajouter un pendule inversé monté en équilibre instable. La rotation infinie du bras assure la stabilisation et le maintien du pendule autour de la verticale au point d'équilibre instable.



Figure 4: Pendule inverse rotatif

4. Quelques applications du pendule inversé

Dans le domaine du transport basé sur l'étude des pendules inversés, on trouve le "Honda Riding Assist", il s'agit d'une moto d'apparence classique, mais qui s'auto équilibre dans les phases critiques (arrêt et basse vitesse). Dépourvue de béquille, conçue pour rester droite à l'arrêt, et pour aider les conducteurs dans les courbes, elle arrive à se maintenir grâce au mouvement de sa fourche.



Figure 5: HONDA RIDING ASSIST-E MOTORCYCLE

Domaine médicale, IBOT, un fauteuil roulant électrique automatisé, qui fonctionne en se basant sur le principe du pendule inversé. C'est un appareil équipé de quatre roues motrices qui est capable d'aider des personnes avec des problèmes de mobilité, de se déplacer en s'assurent un déplacement sur différents types de terrains (trottoirs, escaliers, gravier, etc.) en tout faciliter. Et comme application réussite du principe de ce pendule inverse , on peut citer le Segway et le Hoverboard. C'est deux sont des moyens de transport électrique et individuelle.



Figure 6: HONDA RIDING ASSIST-E MOTORCYCLE



Figure 7: Segway



Figure 8: Hoverboard

III. Conclusion

Nous avons présenté dans ce chapitre quelques types de pendules inverses ainsi que leur domaines d'applications. Dans la suite de ce travail, l'étude se portera sur le pendule inverse classique qui représente un système mécanique très intéressant.

CHAPITRE II

Conception, modélisation et simulation du système

I. Description du système

Notre système est un système multi variable, couplé, non linéaire, instable et sous actionné. Il existe beaucoup d'effets parasites comme le frottement, les modes élastiques de la tige, les effets contrecoup des vitesses etc.

Le but de la commande du pendule inversé est de maintenir en équilibre vertical une tige en aluminium à l'extrémité de laquelle est montée une masse. Cette tige est fixée par une articulation pivotante sur un chariot qui peut se déplacer en glissant le long d'un rail de guidage horizontal.

Le mouvement de rotation d'un moteur électrique est transformé en mouvement de translation du chariot par l'intermédiaire d'une poulie et d'une courroie crantée. Le déplacement du chariot dans un sens ou dans l'autre assure par réaction l'équilibre vertical du bras du pendule. Le pendule est dit inversé car la masse du balancier est au-dessus du chariot.

1) Principe de fonctionnement

Le principe de fonctionnement est très simple en théorie : quand le pendule penche vers la droite, le chariot doit le rattraper en effectuant un mouvement vers la droite et inversement. La difficulté c'est de régler l'intensité et la forme de la réaction du chariot en fonction de l'angle que le pendule fait avec la verticale. Pour contrôler le mouvement du chariot et de l'angle d'inclinaison de la tige nous utilisons capteurs l'un installé sur le moteur et l'autre sur la tige.

II. CONCEPTION AVEC SOLIDWORKS :

On a 5 composants à notre système :

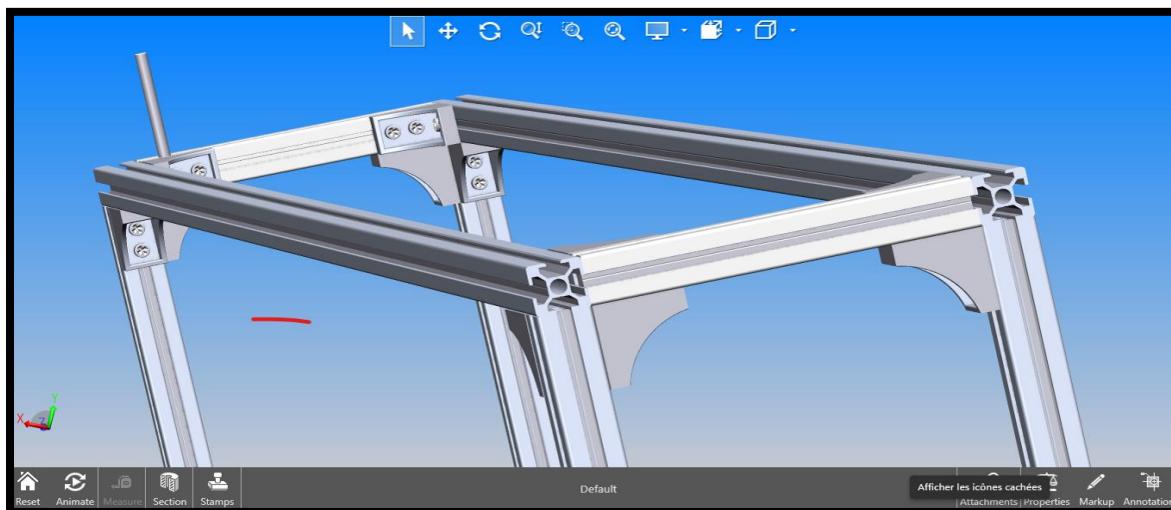


Figure 9: châssis

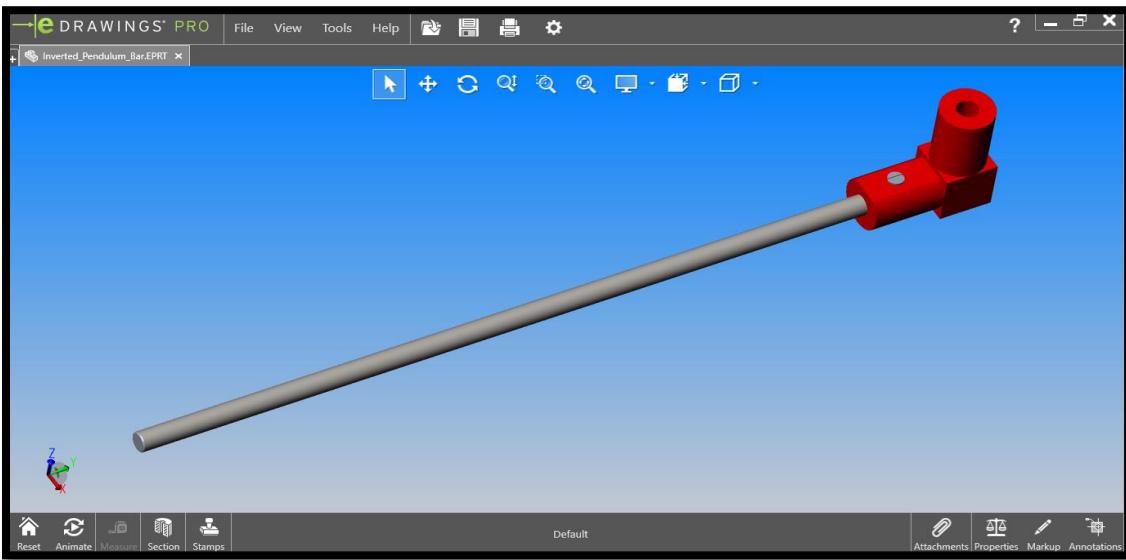


Figure 10: Pendule

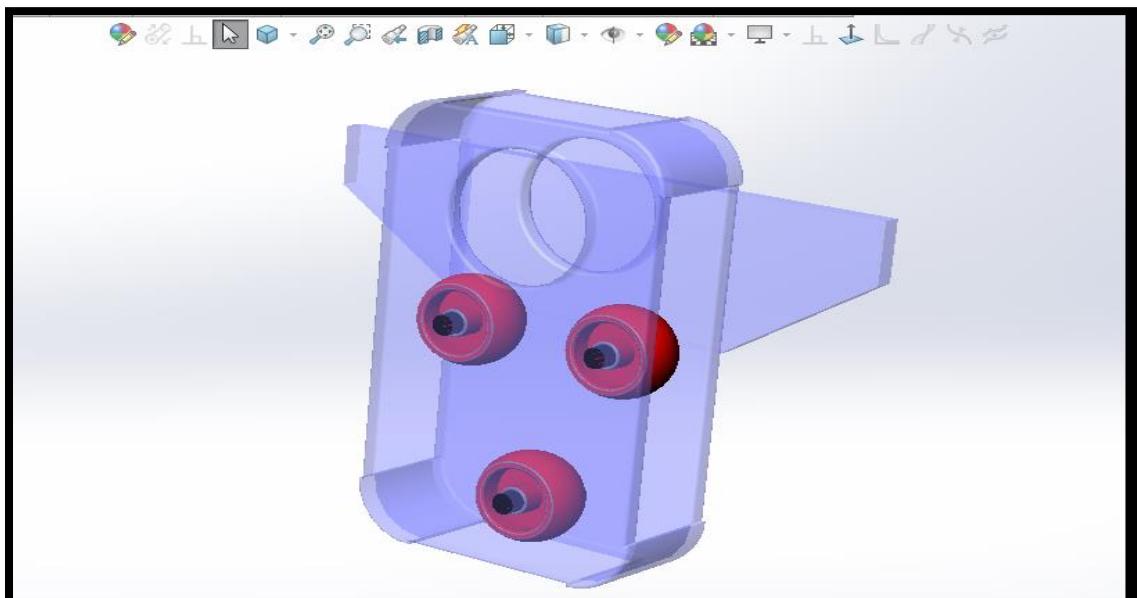


Figure 11: Voiture

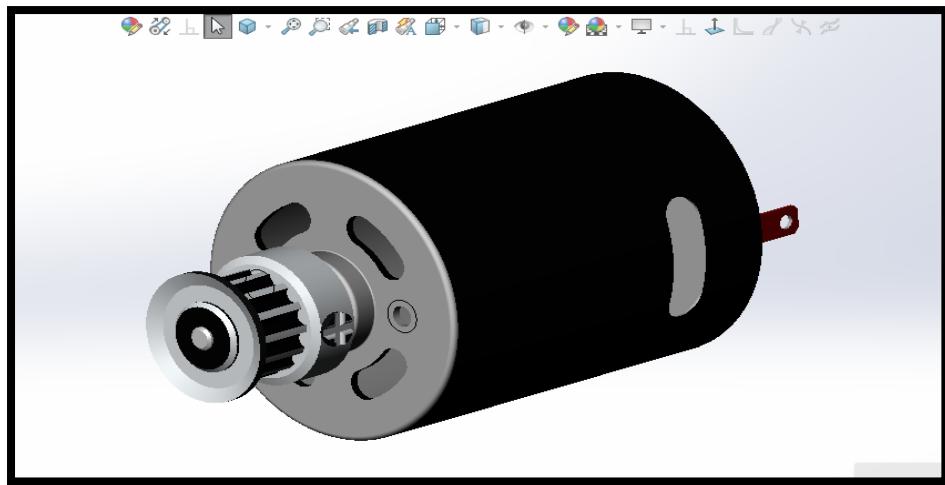


Figure 13: Motor DC avec poulie

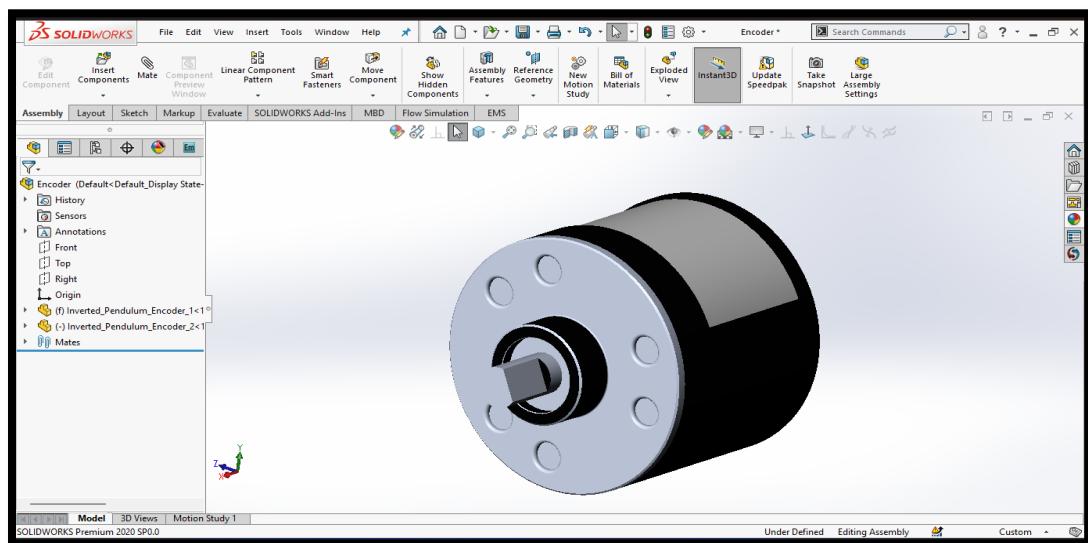


Figure 12: Encodeur

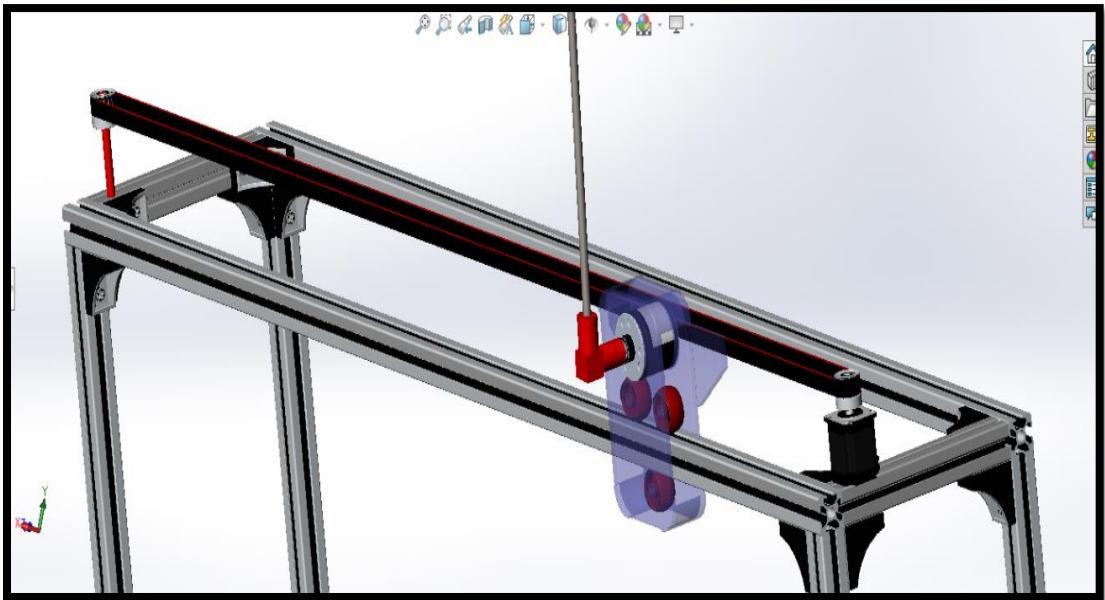


Figure 14: pendule inverse

IV. Modélisation du système (chariot-pendule)

1) Modélisation et régulation d'un pendule inverse :

a) Conception du contrôleur :

✓ Contrôleur PID (Proportionnel, Intégral, Dérivé) :

Le régulateur PID, aussi appelé correct PID (Proportionnel, Intégral, Dérivé) est un système qui permet de contrôler les performances d'un service, qu'il s'agisse d'un système ou d'un appareil installé. Il s'agit de l'appareil le plus utilisé dans l'industrie et la qualité de la correction s'applique à plusieurs physiques majeurs.

⊕ La formule du régulateur PID :

$$u(t) = K_P e(t) + K_I \int_0^t e(t) dt + K_D \frac{de(t)}{dt}$$

Output = Proportional + Integral + Derivative

où :

K_p est le gain proportionnel,

K_i est le gain intégral (on utilise souvent $T_i = 1/K_i$ représentant le temps d'intégration)

K_d est le gain dérivé,

$e(t)$ la fonction d'erreur (consigne - mesure),

$u(t)$ la fonction de commande.

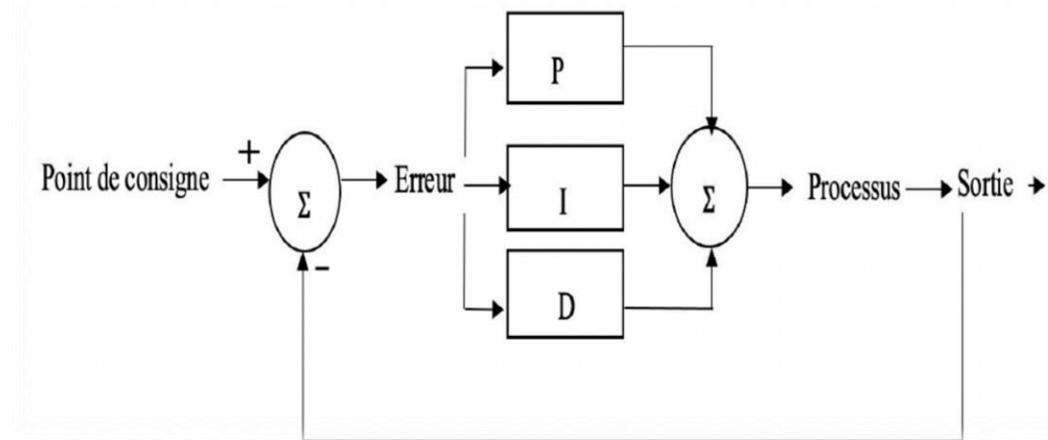


Figure 15: montre le schéma d'un contrôleur PID

Les valeurs que nous avons utilisées pour notre contrôleur ont été trouvées en utilisant l'accordeur manuel offert par la boîte à outils de PID de MATLAB SIMULINK.

✓ **LQR (régulateur linéaire quadratique) :**

Le régulateur quadratique linéaire (LQR) est une méthode bien connue qui fournit des gains de rétroaction contrôlés de manière optimale pour permettre la conception de systèmes stables et hautes performances en boucle fermée.

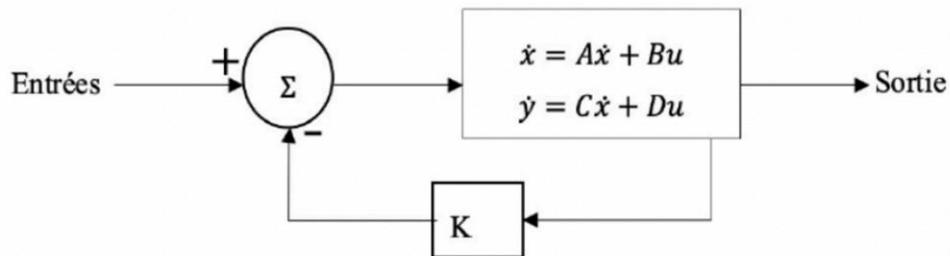
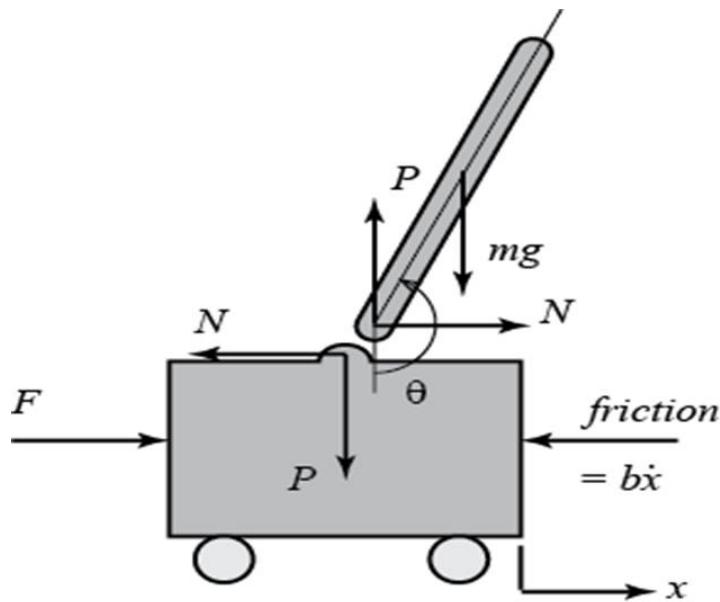


Figure 16: Schéma du contrôleur LQR

Il existe de nombreuses façons de manipuler le comportement d'un système dynamique, et ces approches de contrôle peuvent être trouvées dans la littérature scientifique. Les contrôleurs les plus utilisés sont le LQR et le PID, nous avons donné dans ce chapitre une brève explication sur chaque contrôleur et la façon de l'utiliser. Le prochain chapitre nous verrons la façon dont nous les utiliserons dans notre système et nous donnerons une comparaison entre eux.

b) **Modèle mathématique de pendule inverse :**

Dans cet exemple nous considérerons une version bidimensionnelle du système de pendule inversé avec chariot où le pendule est contraint de se déplacer dans le plan vertical illustré dans la figure ci-dessous. Pour ce système, l'entrée de contrôle est la force F qui déplace et La position horizontale du chariot X .



Pour cet exemple, supposons les quantités suivantes :

- (M) masse du chariot 0,5 kg
- (m) masse du pendule 0,2 kg
- (b) coefficient de frottement pour le chariot 0,1 N/m/sec
- (l) longueur jusqu'au centre de masse du pendule 0,3 m
- (I) moment d'inertie de masse du pendule 0,006 kg.m^2
- (F) force appliquée au chariot
- (x) coordonnées de position du chariot
- (θ) angle du pendule par rapport à la verticale (vers le bas)

Analyse des forces et équations système :

En additionnant les forces dans le diagramme du corps libre du chariot dans la direction horizontale, vous obtenez l'équation de mouvement suivante :

$$M\ddot{x} + b\dot{x} + N = F \quad (1)$$

Notez que vous pouvez également additionner les forces dans la direction verticale pour le chariot, mais aucune information utile ne sera obtenue.

En additionnant les forces dans le diagramme du corps libre du pendule dans la direction horizontale, vous obtenez l'expression suivante pour la force de réaction N :

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \quad (2)$$

Si vous remplacez cette équation dans la première équation, vous obtenez l'une des deux équations régissant ce système :

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F \quad (3)$$

Pour obtenir la deuxième équation du mouvement de ce système, additionnez les forces perpendiculaires au pendule. Résoudre le système le long de cet axe simplifie grandement les mathématiques. Vous devriez obtenir l'équation suivante :

$$P \sin \theta + N \cos \theta - mg \sin \theta = ml\ddot{\theta} + m\ddot{x} \cos \theta \quad (4)$$

Pour supprimer les termes \mathbf{P} et \mathbf{N} dans l'équation ci-dessus, additionnez les moments autour du centre de gravité du pendule pour obtenir l'équation suivante.

$$-Pl \sin \theta - Nl \cos \theta = I\ddot{\theta} \quad (5)$$

En combinant ces deux dernières expressions, vous obtenez la deuxième équation directrice :

$$(I + ml^2)\ddot{\theta} + mgl \sin \theta = -ml\ddot{x} \cos \theta \quad (6)$$

Étant donné que les techniques d'analyse et de conception de contrôle que nous utiliserons dans cet exemple s'appliquent uniquement aux systèmes linéaires, cet ensemble d'équations doit être linéarisé. Plus précisément, nous linéariserons les équations sur la position d'équilibre verticalement ascendante, $\theta = \pi$, et supposerons que le système reste dans un petit voisinage de cet équilibre. Cette hypothèse devrait être raisonnablement valable puisque sous contrôle, nous souhaitons que le pendule ne s'écarte pas de plus de 20 degrés de la position verticale vers le haut. Soit ϕ représente l'écart de la position du pendule par rapport à l'équilibre, c'est-à-dire $\theta = \pi + \phi$. En supposant encore une fois un petit écart (ϕ) par rapport à l'équilibre, nous pouvons utiliser les approximations aux petits angles suivantes des fonctions non linéaires dans nos équations système :

$$\cos \theta = \cos(\pi + \phi) \approx -1 \quad (7)$$

$$\sin \theta = \sin(\pi + \phi) \approx -\phi \quad (8)$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0 \quad (9)$$

Après avoir remplacé les approximations ci-dessus dans nos équations directrices non linéaires, nous arrivons aux deux équations linéarisées du mouvement. Notez que \mathbf{u} a été remplacé par l'entrée \mathbf{F} .

$$(I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \quad (10)$$

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \quad (11)$$

Fonction de transfert :

Pour obtenir les fonctions de transfert des équations du système linéarisées, il faut d'abord prendre la transformée de Laplace des équations du système en supposant des conditions initiales nulles. Les transformées de Laplace résultantes sont présentées ci-dessous :

$$(I + ml^2)\Phi(s)s^2 - mgl\Phi(s) = mlX(s)s^2 \quad (12)$$

$$(M + m)X(s)s^2 + bX(s)s - ml\Phi(s)s^2 = U(s) \quad (13)$$

Rappelons qu'une fonction de transfert représente la relation entre une seule entrée et une seule sortie à la fois. Pour trouver notre première fonction de transfert pour la sortie $\Phi(s)$ et une entrée de $\mathbf{U}(s)$, nous devons éliminer $\mathbf{X}(s)$ des équations ci-dessus. Résolvez la première équation pour $\mathbf{X}(s)$:

$$X(s) = \left[\frac{I + ml^2}{ml} - \frac{g}{s^2} \right] \Phi(s) \quad (14)$$

Remplacez ensuite ce qui précède dans la deuxième équation :

$$(M + m) \left[\frac{I + ml^2}{ml} - \frac{g}{s^2} \right] \Phi(s)s^2 + b \left[\frac{I + ml^2}{ml} - \frac{g}{s^2} \right] \Phi(s)s - ml\Phi(s)s^2 = U(s) \quad (15)$$

En réorganisant, la fonction de transfert est alors la suivante :

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s^2}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s} \quad (16)$$

$$\text{Où : } q = [(M + m)(I + ml^2) - (ml)^2] \quad (17)$$

D'après la fonction de transfert ci-dessus, on peut voir qu'il y a à la fois un pôle et un zéro à l'origine. Ceux-ci peuvent être annulés et la fonction de transfert devient la suivante :

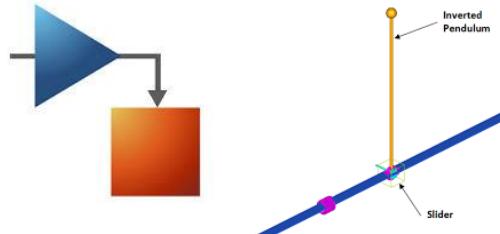
$$P_{pend}(s) = \frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmgl}{q}} \quad \left[\frac{rad}{N} \right] \quad (18)$$

Deuxièmement, la fonction de transfert avec la position du chariot $\mathbf{X}(s)$ comme sortie peut être dérivée de la même manière pour arriver à ce qui suit :

$$P_{cart}(s) = \frac{X(s)}{U(s)} = \frac{\frac{(I+ml^2)s^2 - gml}{q}}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s} \quad \left[\frac{m}{N} \right]$$

3) Modélisation par Matlab Simulink (Schéma Block) :

i. Construire le modèle non linéaire avec MATLAB - Simulink



Dans cette section, nous montrons alternativement comment construire le modèle de pendule inversé en utilisant les blocs de modélisation physique de l'extension Simscape de Simulink. Les blocs de la bibliothèque Simscape représentent des composants physiques réels ; par conséquent, des modèles dynamiques multi-corps complexes peuvent être construits sans qu'il soit nécessaire de construire des équations mathématiques à partir de principes physiques comme cela a été fait ci-dessus en appliquant les lois de Newton.

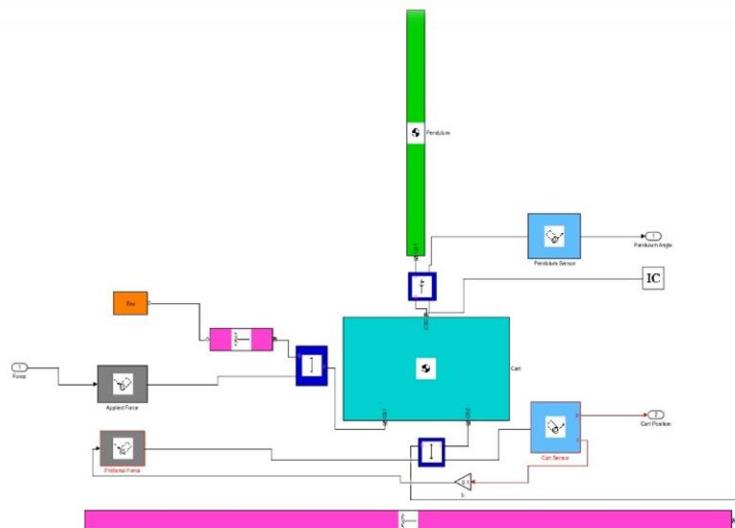


Figure 17: pendule inverse dans SIMSCAPE

ii. Pour faire cela, il faut suivre des étapes :

- ❖ Insérez un bloc Carrosserie de la bibliothèque Simscape/Multi-body/First Génération(1G)/Bodies pour représenter chariot :

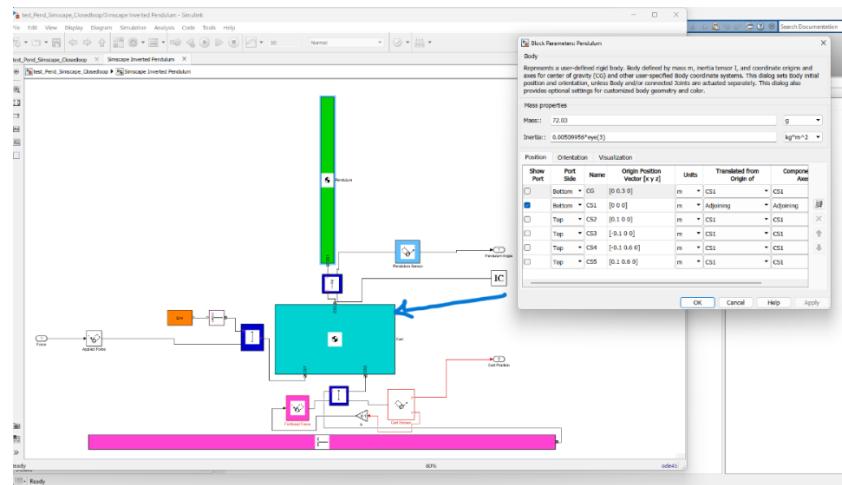


Figure 18: le bloc du chariot

- ❖ Insérez un deuxième bloc Corps pour représenter le pendule :

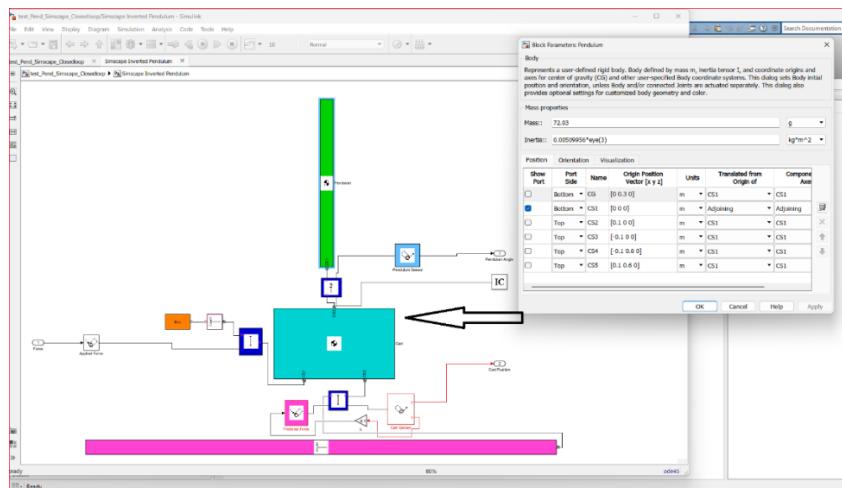


Figure 19: le bloc du pendule

- ❖ Ajoutez ensuite un bloc conjoint pour définir l'articulation dépendante du pendule au chariot, et pour le contact de la force avec le chariot, ainsi que pour les efforts de friction avec l'air :

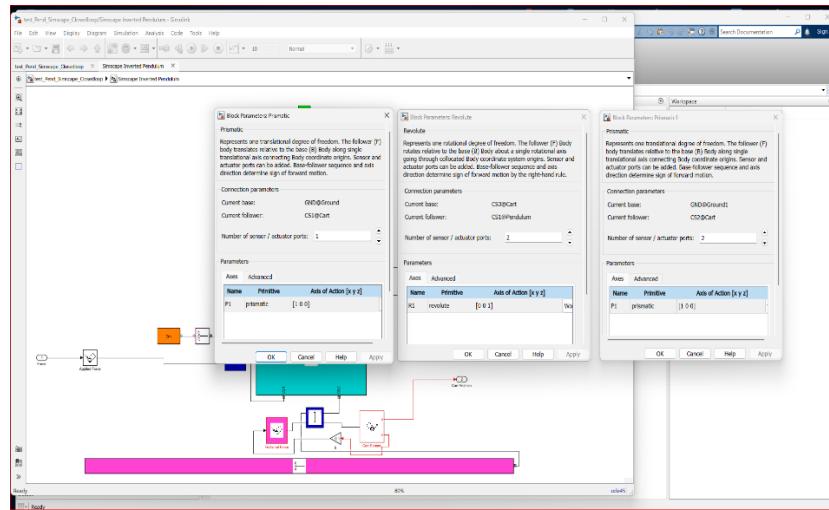


Figure 20: le bloc des joints

- ❖ On ajoute les capteurs :

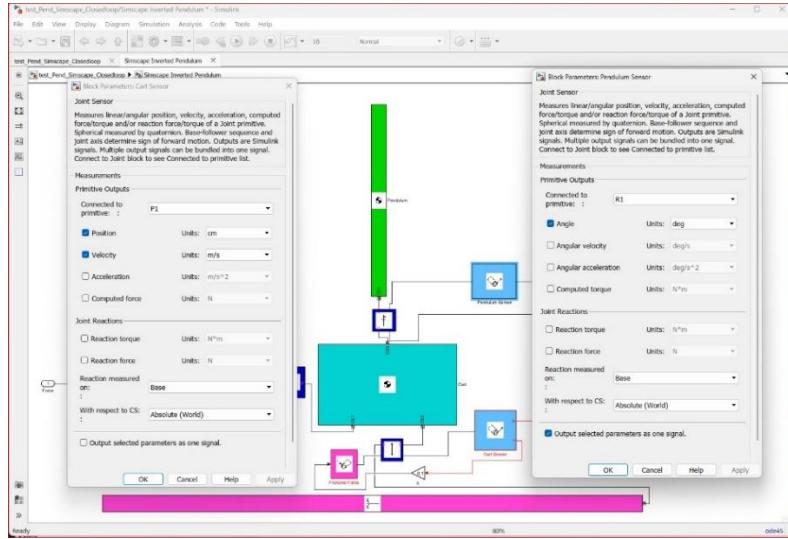


Figure 21: le bloc des capteurs

- ❖ Pour ajouter les actionneurs :

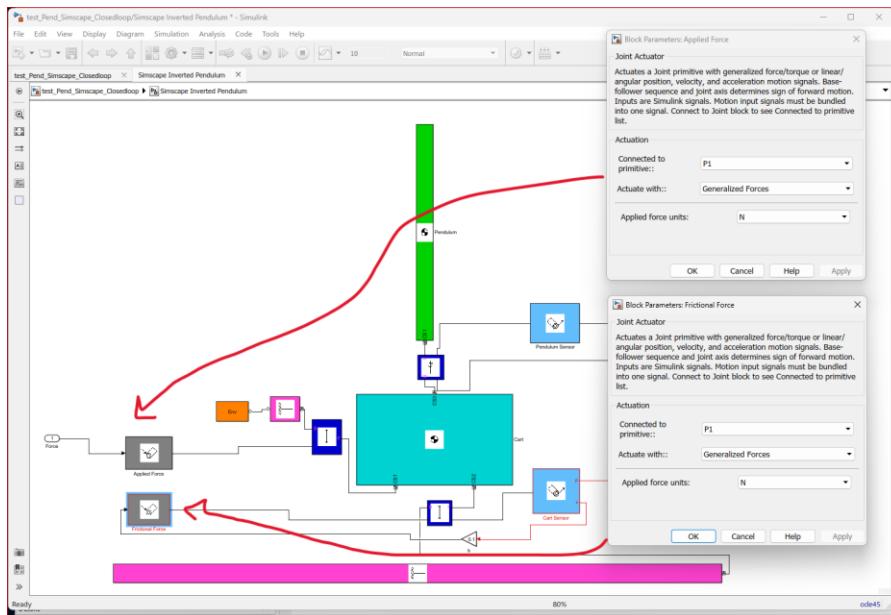


Figure 22: le bloc des actionneurs

- ❖ Pour ajouter l'environnement et les bases :

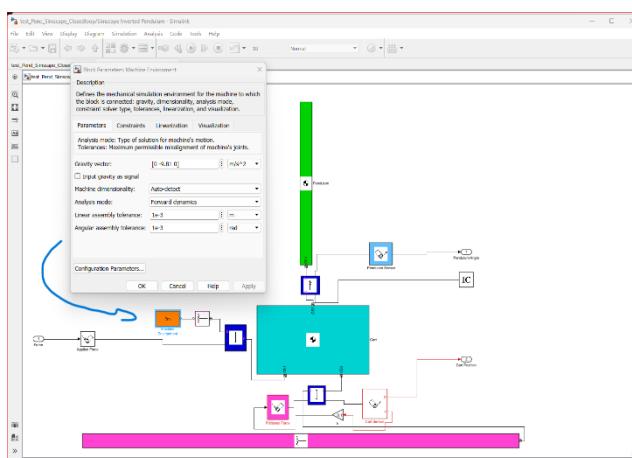


Figure 23: le bloc de l'environnement

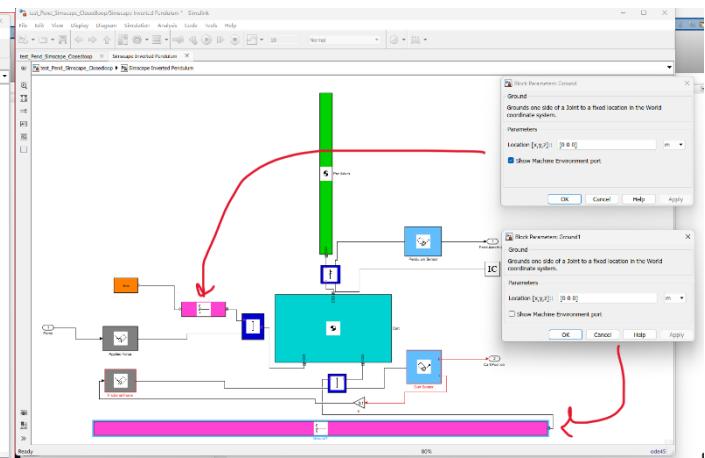


Figure 24 : le bloc des bases

- ❖ Enfin, ajoutez la perturbation extérieure modélisée par une force, ainsi que les oscilloscopes pour la visualisation du comportement de notre pendule :

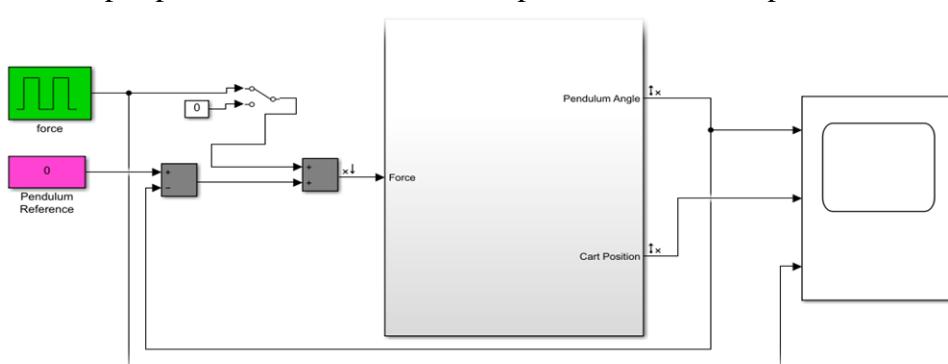


Figure 25: pendule inverse sans PID

- ❖ Ensuite, ouvrez le Scope. Vous verrez le résultat suivant pour l'angle du pendule et la position du chariot.

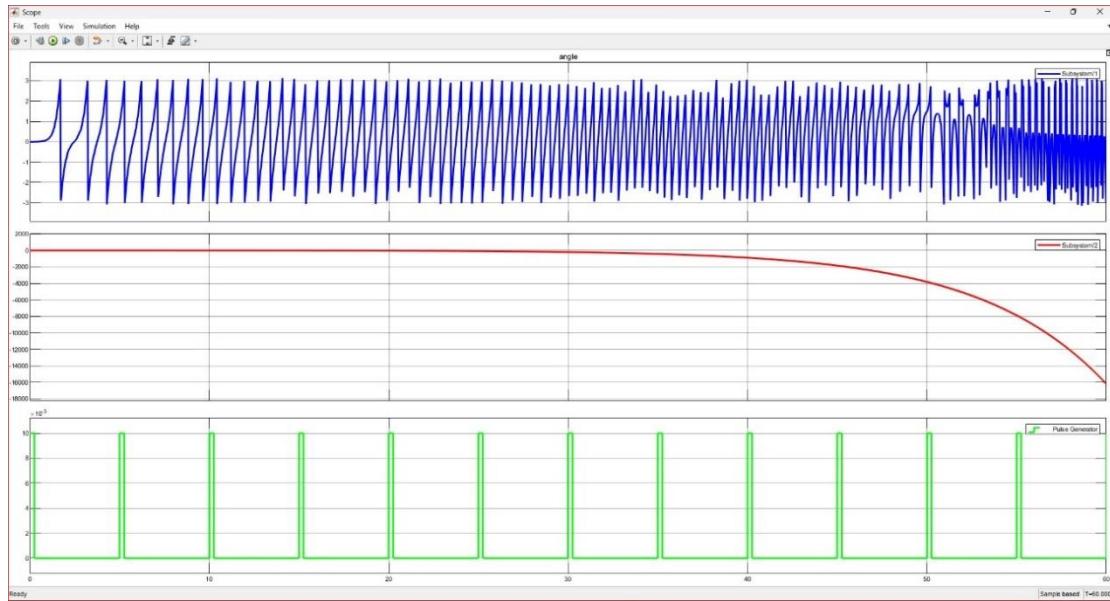


Figure 26 : le résultat

Notez que le pendule effectue à plusieurs reprises des révolutions complètes où l'angle roule à $-\pi$ et π radians. De plus, la position du chariot s'étend sans limite, mais oscille sous l'influence du pendule oscillant.

- ❖ Pour réaliser la stabilisation du système, il est nécessaire d'ajouter le système de contrôle PID :

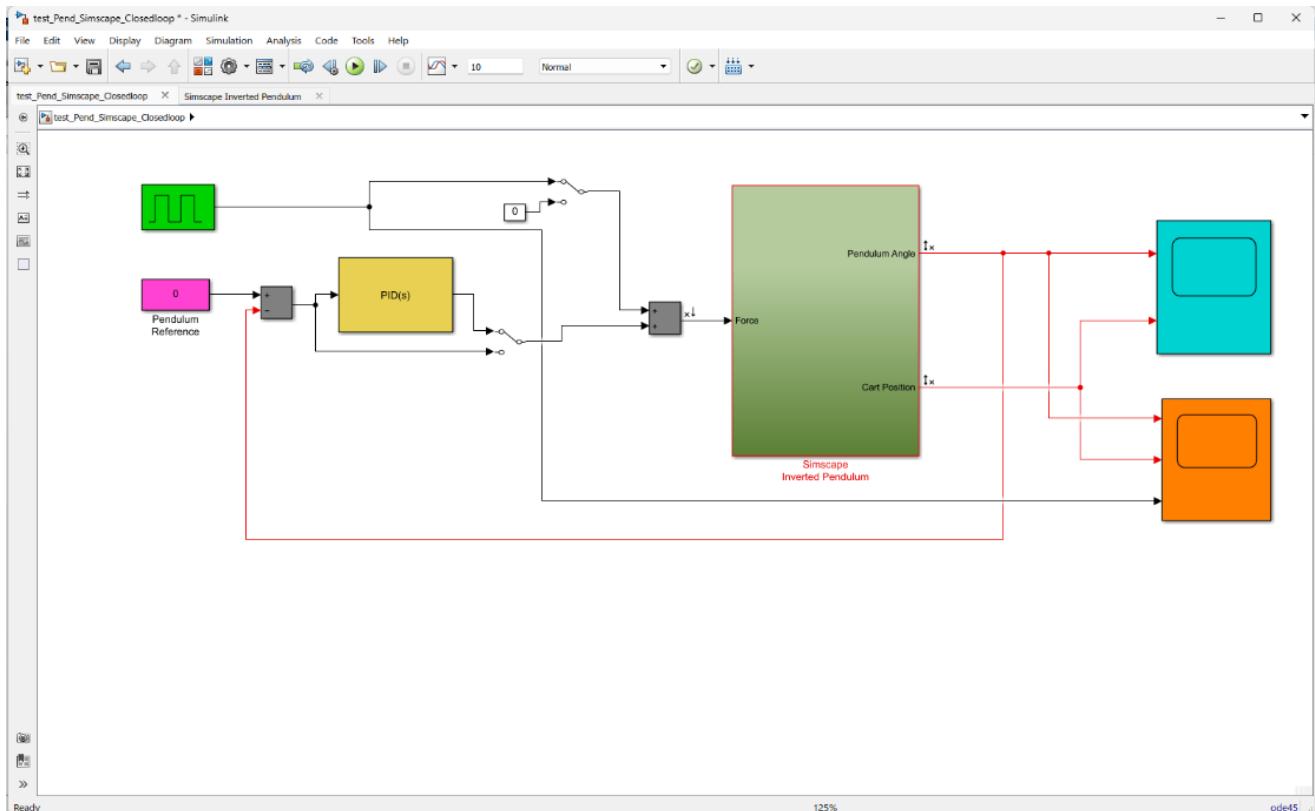


Figure 27: pendule inverse avec PID

- ❖ Et effectuer l'opération de "tuning" pour trouver les valeurs adéquates de KP, KI, KD pour le PID, permettant ainsi de stabiliser notre système :

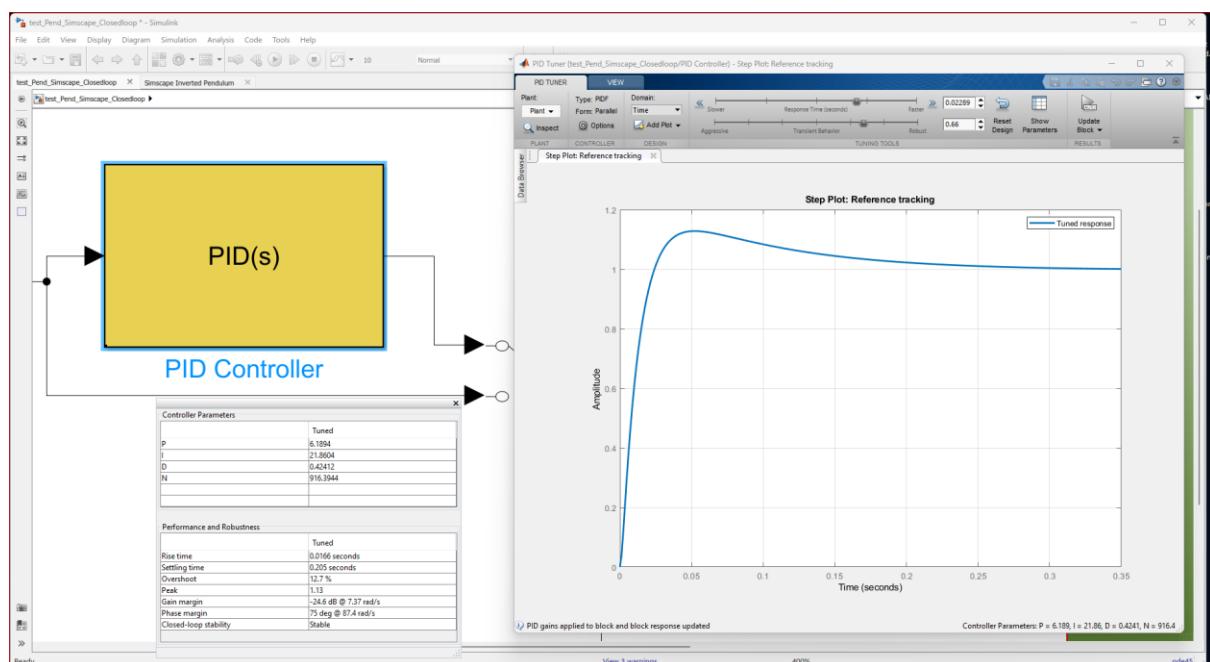


Figure 28 : l'opération de "tuning"

- ❖ Suite à cette opération, nous procédon à la re-simulation du système :

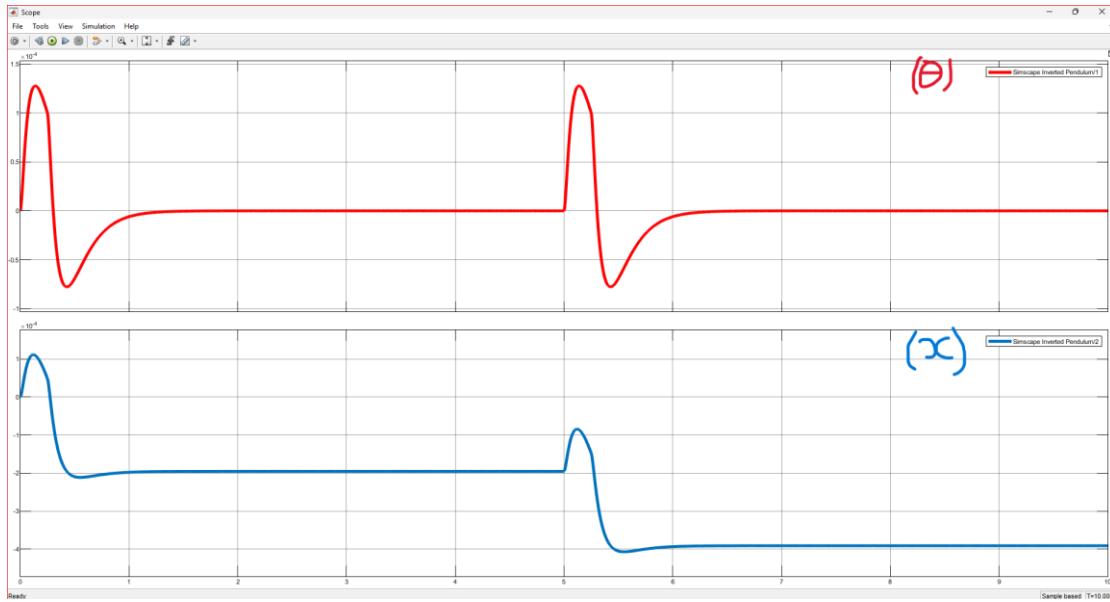


Figure 29: le résultat

L'exécution d'une simulation produit les deux tracés suivants montrant la réponse contrôlée du système. Après l'impact initial, le contrôleur a pu rapidement ramener l'angle du pendule à zéro et la vitesse du pendule est également nulle. Le chariot se déplace lentement et avec une vitesse constante dans la direction X négative pour maintenir l'équilibre du pendule.

- ❖ extraire la fonction de transfert d'un système à partir d'une simulation :

Lorsque vous travaillez avec Simulink pour simuler un système physique, il est souvent nécessaire de linéariser le système pour obtenir sa fonction de transfert. La linéarisation consiste à rapprocher le comportement d'un système non linéaire autour d'un point de fonctionnement spécifique :

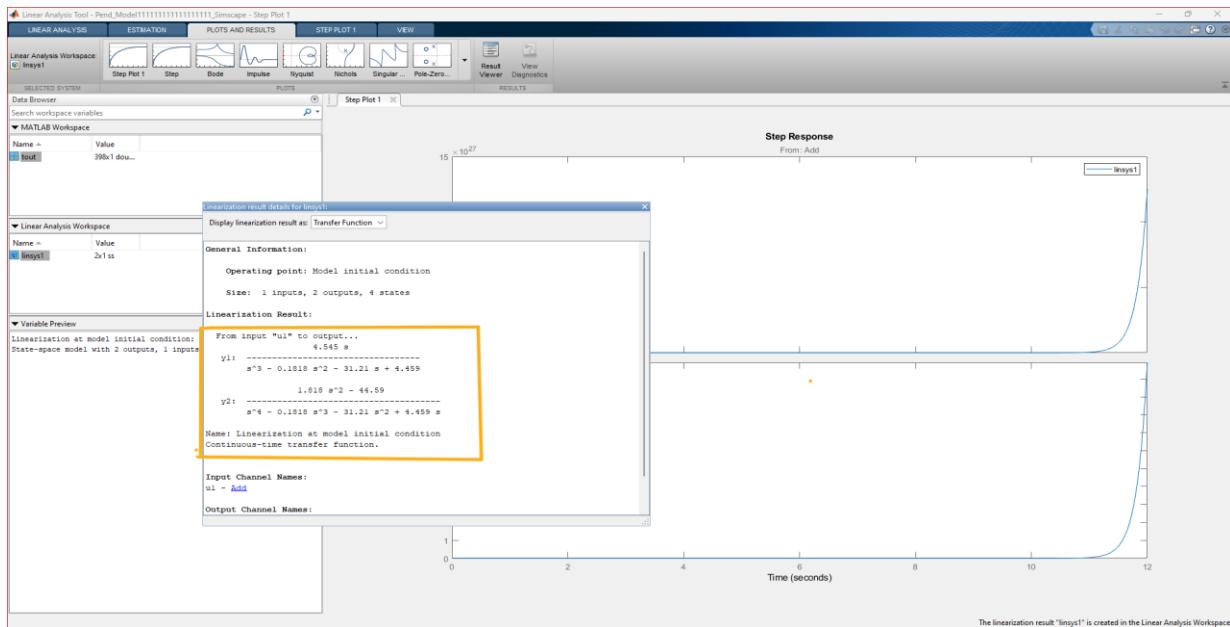
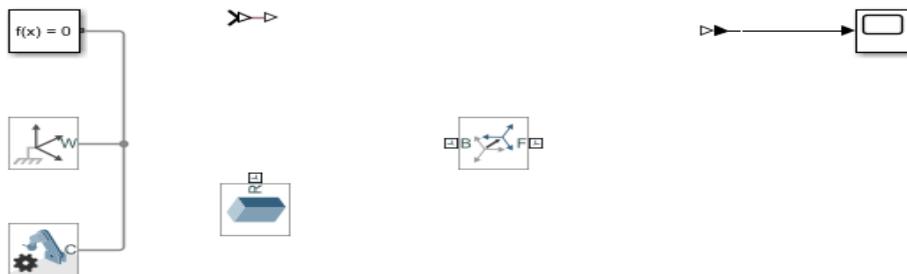


Figure 30: la fonction de Transfer

i. Pendule inversé : modélisation Simscape Configuration physique

Dans cette section, nous montrons comment construire le modèle de pendule inversé à l'aide des blocs de modélisation physique de Simscape Multi-body. Les blocs de la bibliothèque Simscape représentent des composants physiques réels ; par conséquent, des modèles dynamiques multicorps complexes peuvent être construits sans avoir besoin de construire des équations mathématiques à partir de principes physiques, comme cela a été fait en appliquant les lois de Newton pour générer le modèle implémenté dans Pendule inversé : Modélisation Simulink.

➤ Créez un cadre mondial et une configuration de base :



Simscape Multibody Resources

1. Find more multibody components in the [Simscape Multibody library](#).
For more information, see [Simscape Multibody - Blocks](#).
2. Find components from other domains in the [Simscape library](#).
3. Connect the components to form a physical network.
For more information, see [Essential Steps for Constructing a Physical Model](#) and [Creating a Multibody Model](#).
4. Visualize the simulation using [Mechanics Explorer](#)
5. [Explore simulation results](#) using `sscexplore`

Figure 31: un cadre mondial

- Assembler le plan de base et le chariot et ajoute L'articulation prismatique sera actionnée par une entrée de force :

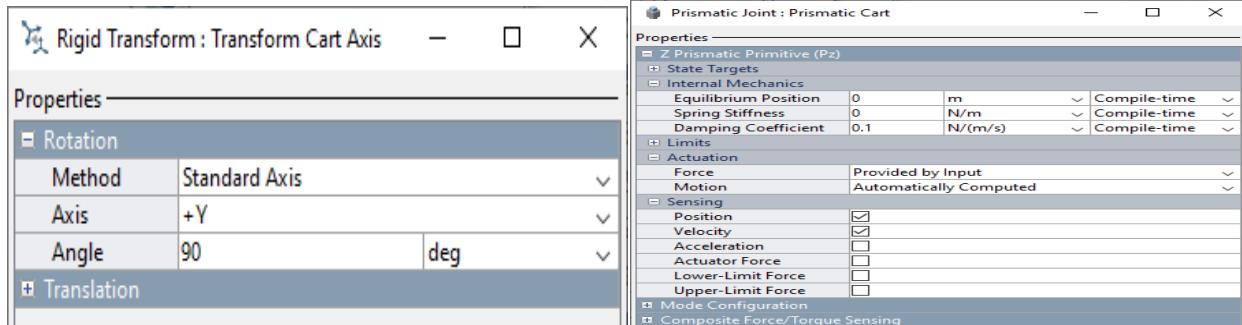


Figure 32: le bloc du joint de translation

- Puisque nous modélisons le chariot coulissant comme une masse ponctuelle, seule la masse affectera les résultats de la simulation :

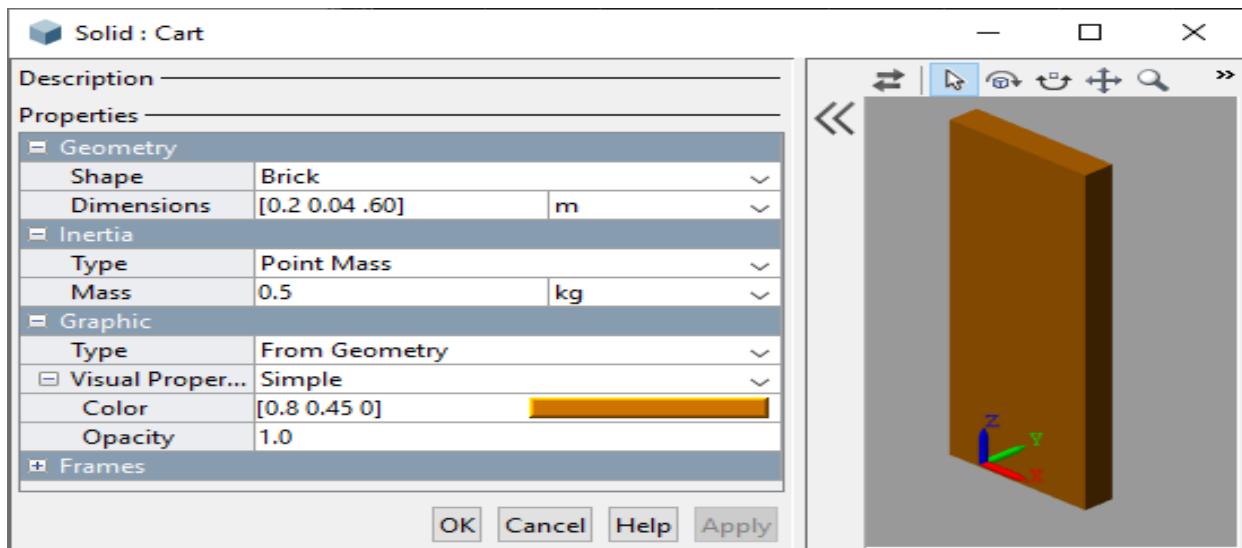


Figure 33: le bloc du chariot

- Le modèle à ce stade devrait maintenant apparaître comme suit :

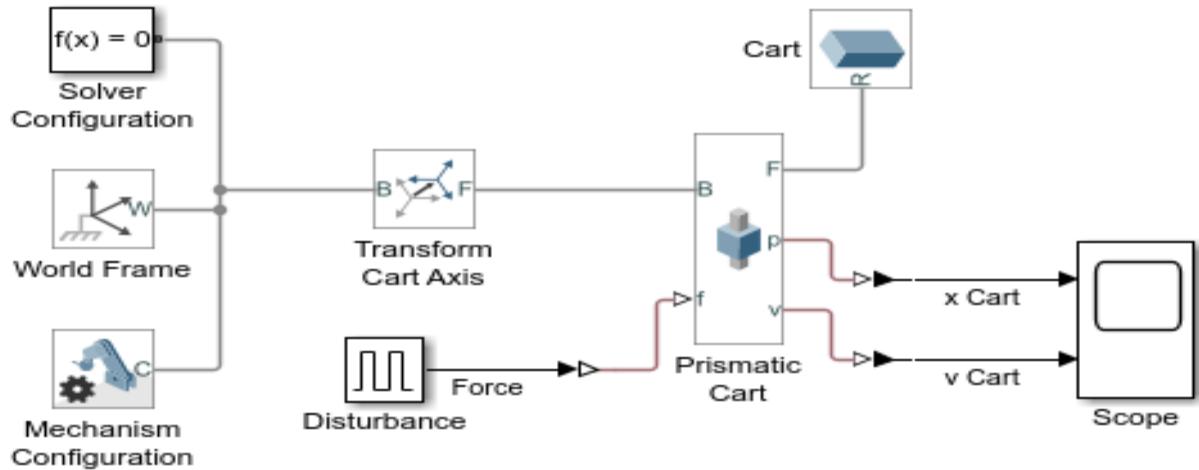


Figure 34: le chariot dans de la pendule dans simscape

- Sous-système pendulaire et connexion du chariot au pendule :

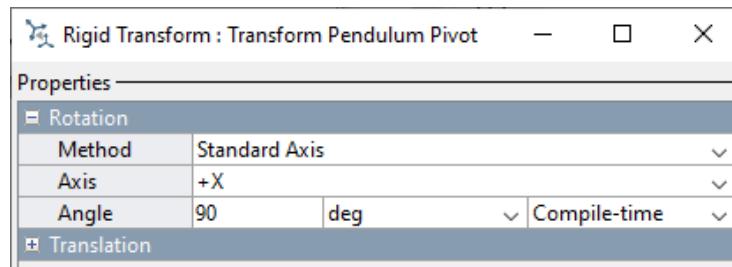


Figure 35: l'articulation pivot

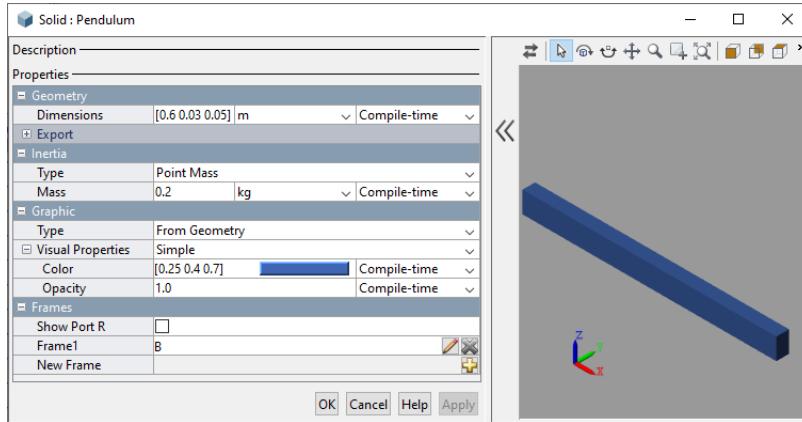


Figure 36 : le bloc de pendule

Le modèle résultant devrait apparaître comme suit :

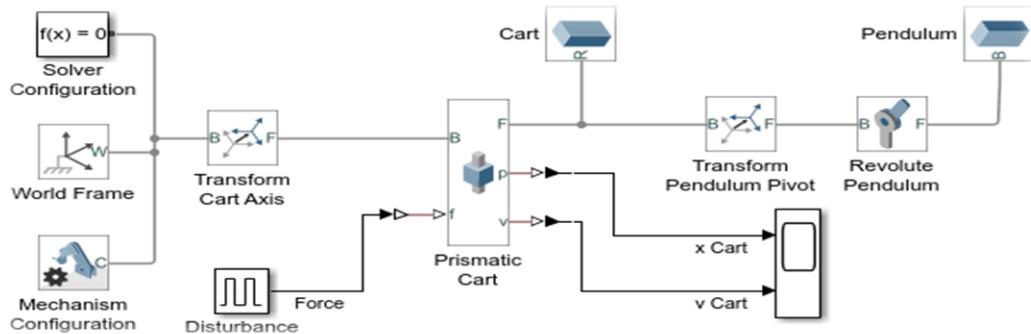


Figure 37: le schéma physique de pendule inverse dans Simscape

En exécutant une simulation, le tracé suivant est généré, où l'on peut voir que l'ajout du pendule modifie le comportement du chariot à la fois dans sa distance parcourue ainsi que dans sa vitesse.



Figure 38: le résultat du simulation sans PID

- Et lorsque on ajoute le contrôleur PID :

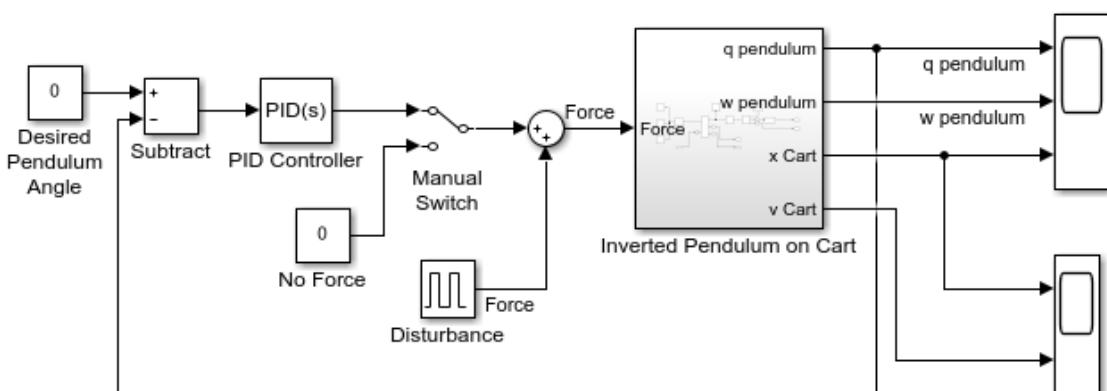


Figure 39: pendule inverse avec le contrôleur PID

En trouve le résultat suivant :

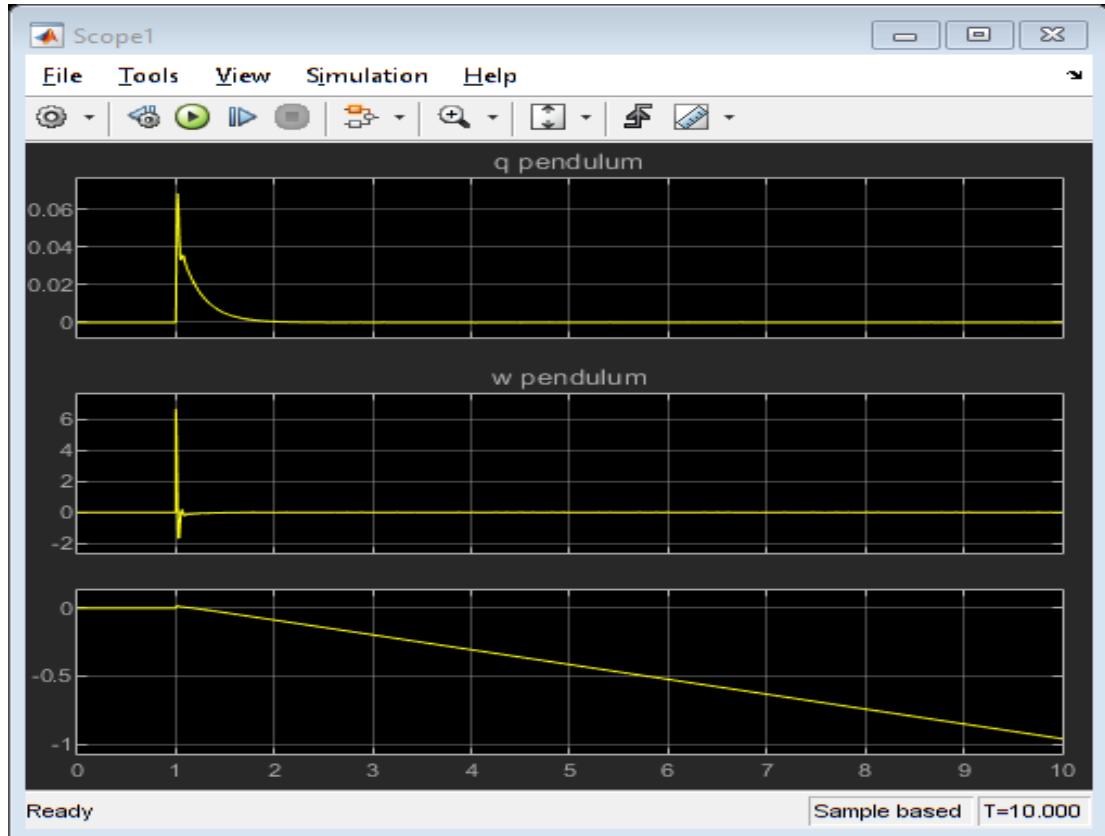


Figure 40: le résultat du simulation avec PID

Les résultats de simulation confirment l'importance fondamentale du régulateur PID dans la stabilisation et la contrôle du système pendulaire inverse. Il ajuste la proportion d'erreurs, intègre les erreurs accumulées et protège contre les tendances brusques du variateur, le PID est presque toujours une prise indispensable pour entretenir l'adaptateur pendulaire. Des ajustements dynamiques sont assurés par le PID sur l'affichage des effets du système non linéaire, pour autant que le pendule soit essentiel au fonctionnement principal du pendule en marche arrière.

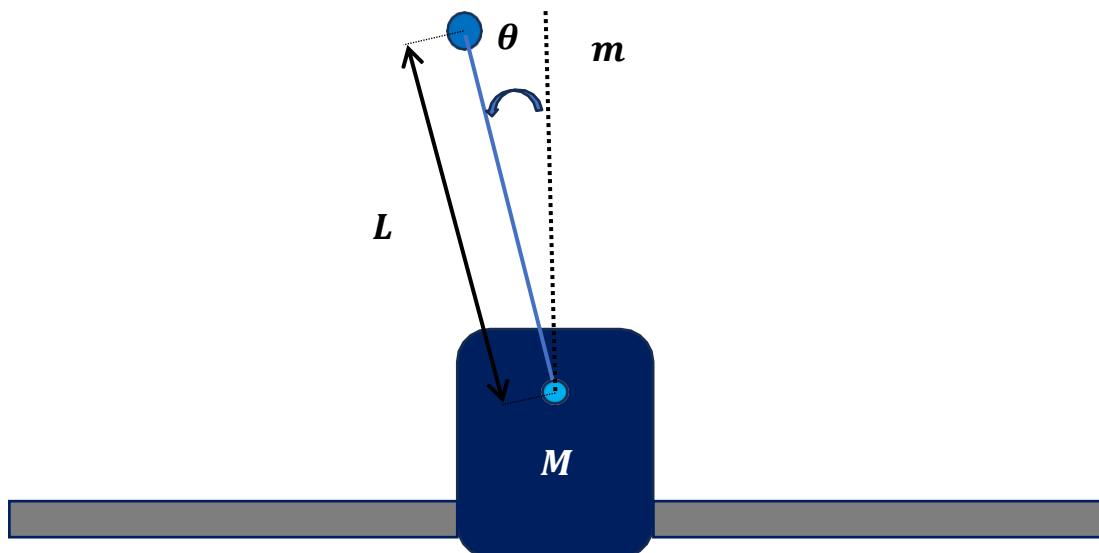
4) Modélisation par 20-Sim :

20-sim est un logiciel de modélisation et de simulation largement utilisé pour la conception et l'analyse de systèmes dynamiques complexes. Offrant une interface graphique intuitive, il permet aux ingénieurs et aux chercheurs de représenter visuellement les composants d'un système et de définir leurs interactions à l'aide de schémas blocs. Ce logiciel prend en charge la modélisation multidisciplinaire, ce qui signifie qu'il peut être appliqué à une variété de domaines tels que la mécanique, l'électrique et l'hydraulique.

Dans le contexte de la modélisation d'un pendule inversé, l'utilisateur peut décrire les équations du mouvement dérivées de la mécanique newtonienne ou lagrangienne. Ces équations sont ensuite traduites en un schéma bloc dans 20-sim, où chaque composant du système est représenté par un bloc interconnecté. Des paramètres tels que la longueur du pendule, la masse, et les propriétés du moteur peuvent être définis pour refléter les caractéristiques spécifiques du système physique.

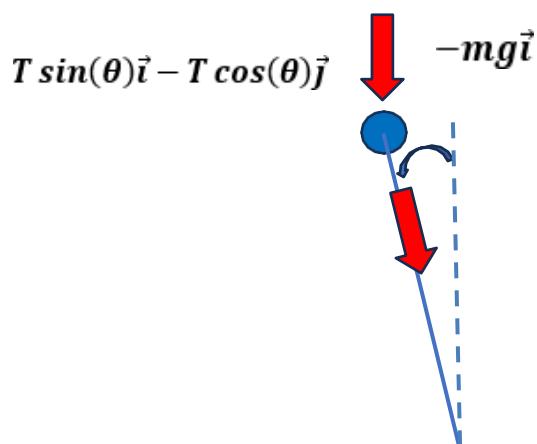
La simulation dans 20-sim offre la possibilité de visualiser le comportement du pendule inversé en réponse à différentes entrées, permettant ainsi aux ingénieurs de comprendre et d'analyser le système dans des conditions diverses.

✓ Génération des équations par la mécanique newtonienne

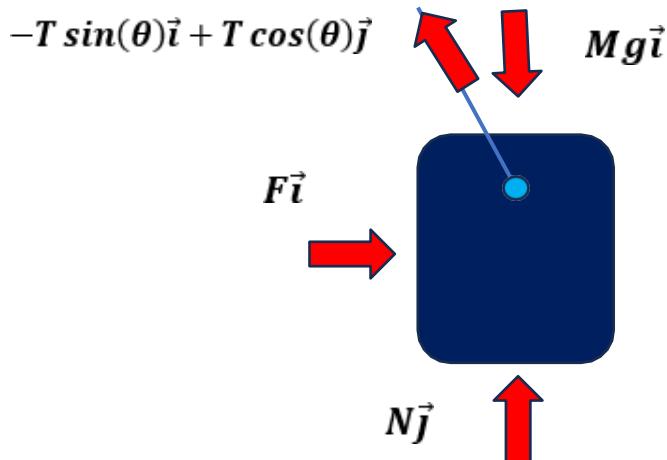


➤ Extraction des forces appliquées sur le pendule et le Car :

- Extraction des forces appliquées sur le pendule :



- Extraction des forces appliquées sur le Car :



- Application de la loi de Newton :

- Sur le Car :

$$\hat{l}: F - T \sin(\theta) = M\ddot{x} \quad 1$$

- Sur le Pendule :

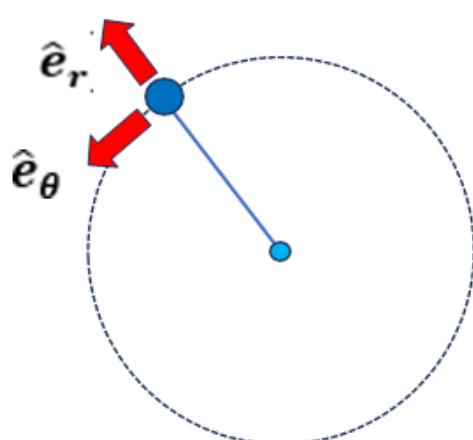
$$\hat{l}: T \sin(\theta) = m \cdot A_{Px} \quad 2$$

$$\hat{j}: T \cos(\theta) - mg = m \cdot A_{Py} \quad 3$$

Avec :

A_{Px} Est l'accélération du pendule suivant l'axe (Ox)

A_{Py} Est l'accélération du pendule suivant l'axe (Oy)



On a :

$$\mathbf{A}_p = \mathbf{A}_C + \mathbf{A}_{P/C}$$

\mathbf{A}_p Est l'accélération du pendule

Avec : \mathbf{A}_C Est l'accélération du Car

$\mathbf{A}_{P/C}$ Est l'accélération du pendule par rapport au repère du Car

Donc :

$$\begin{aligned} \mathbf{A}_p &= \ddot{x}\hat{i} + [L\ddot{\theta}\hat{e}_\theta - L\dot{\theta}^2\hat{e}_r] \\ &= \ddot{x}\hat{i} + L\ddot{\theta}[-\cos(\theta)\hat{i} - \sin(\theta)\hat{j}] \\ &= -L\dot{\theta}^2[\sin(\theta)\hat{i} + \cos(\theta)\hat{j}] \end{aligned}$$

2 \rightarrow $T \sin(\theta) = m\ddot{x} - mL\ddot{\theta} \cos(\theta) + mL\dot{\theta}^2 \sin(\theta)$ 4
 3 \rightarrow $-T \cos(\theta) - mg = -mL\ddot{\theta} \sin(\theta) - mL\dot{\theta}^2 \cos(\theta)$ 5

4 $\cos(\theta) +$ 5 $\sin(\theta)$
 ↓
 -mg sin(θ) = mddot{x} cos(θ) - mLddot{θ} 6
 F + mLddot{θ} cos(θ) - mLdot{θ}² sin(θ) = (m+M)ddot{x} 7

L'angle (θ) est proche de 0. Alors on considère les approximations suivantes :

$$\sin(\theta) = \theta$$

$$\cos(\theta) = 1$$

Alors :

6 \rightarrow $-mg\theta = m\ddot{x} - mL\ddot{\theta}$ 8
 7 \rightarrow $F + mL\ddot{\theta} - mL\dot{\theta}^2\theta = (m+M)\ddot{x}$ 9

Finalement :

8 \rightarrow $\ddot{x} = k_1\dot{\theta}^2\theta + k_2\theta + k_3$

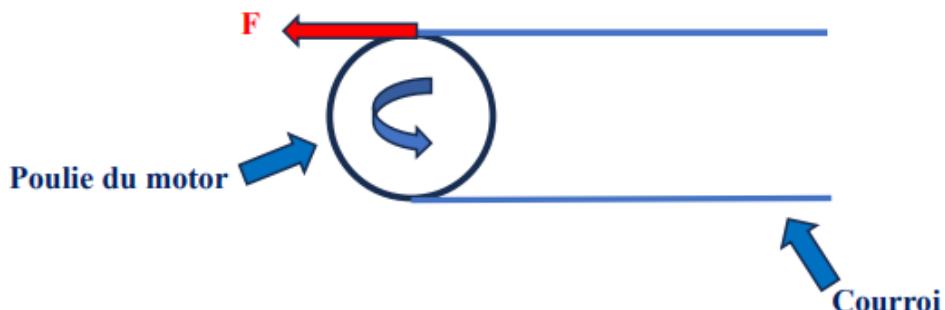
Avec :

$$k_1 = \frac{-mL}{M}$$

$$k_2 = \frac{mg}{M}$$

$$k_3 = \frac{F}{M}$$

F est la force appliquée par le moteur sur le courroï



On a :

$$c = k \cdot i$$

$$c = R \cdot F$$

Alors :

$$F = \frac{k \cdot i}{R}$$

Avec :

- K est une caractéristique du motor
- C est le couple du motor
- R est le rayon du poulie du motor

✓ Réalisation du schéma block dans l'environnement de 20-Sim :

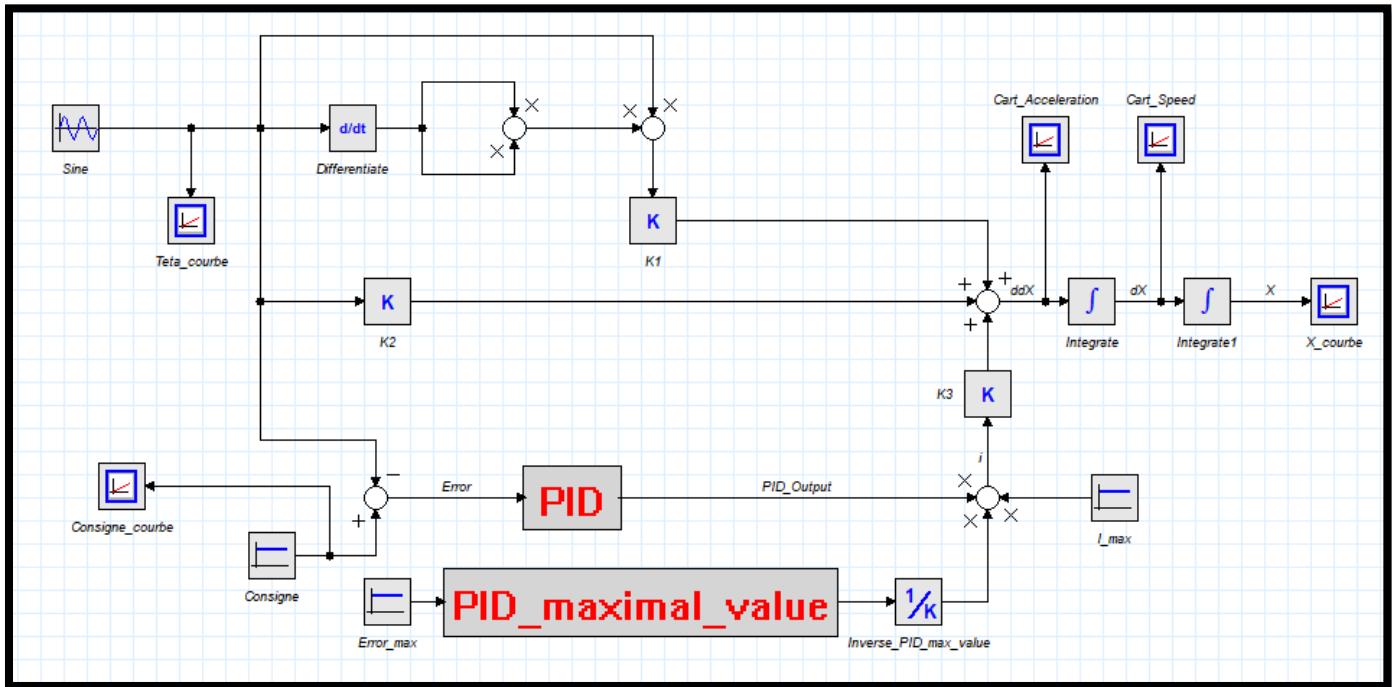


Figure 41: Schéma block du système

Ce schéma bloc m'a permis de traduire les équations du mouvement en un modèle visuel, facilitant ainsi la compréhension de la dynamique complexe du pendule inversé.

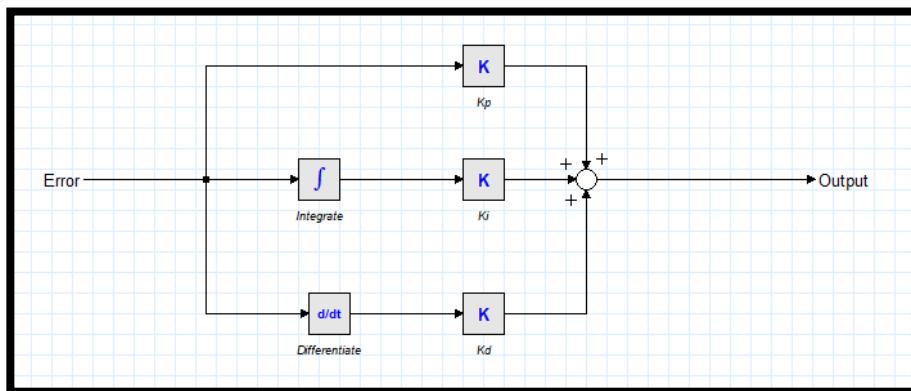


Figure 42: PID_Sous_Block

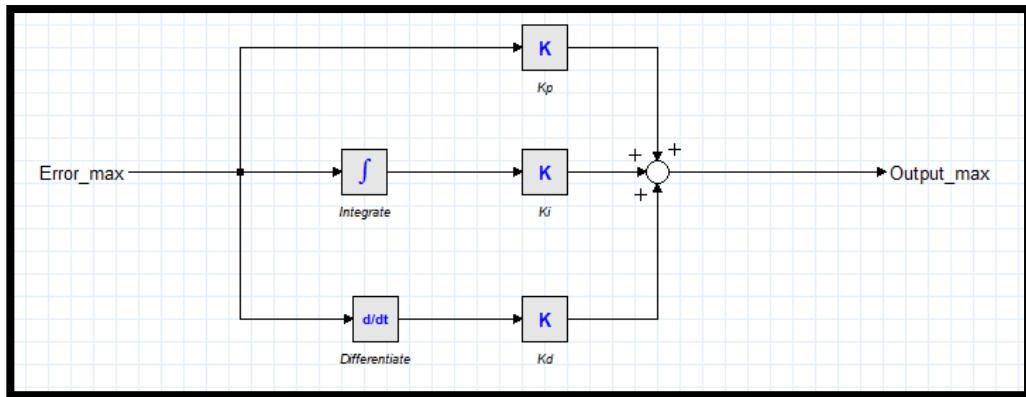


Figure 43: PID_Maximal_Value_Sous_Block

✓ Génération des courbes :

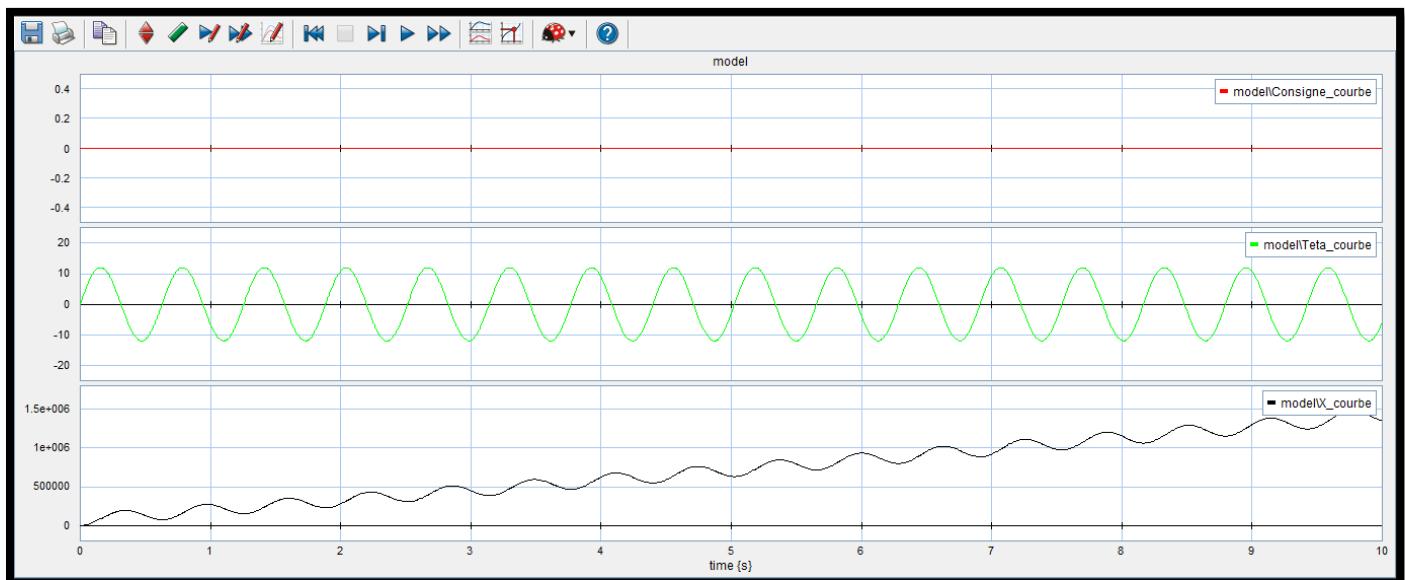


Figure 44: Variation de l'angle Théta / Déplacement du chariot

Les résultats de la simulation, où nous avons appliqué un signal sinusoïdal à l'angle theta du pendule inversé, ont clairement démontré la relation dynamique entre l'angle du pendule et le mouvement du chariot. Comme anticipé, le chariot a commencé à se déplacer de manière sinusoïdale en réponse aux oscillations du pendule. Cette observation confirme la nature interdépendante du système, où les variations subtiles de l'angle du pendule déclenchent des réponses significatives au niveau du chariot. La périodicité du signal sinusoïdal à l'angle theta se reflète directement dans le déplacement sinusoïdal du chariot, illustrant la manière dont le système réagit de manière cohérente et synchronisée.

Ces résultats soulignent l'importance de la coordination entre les composants du système pour maintenir l'équilibre du pendule. La simulation avec un signal sinusoïdal offre une première approche pour comprendre le comportement du système dans des conditions spécifiques, et cette observation encourage à explorer davantage les caractéristiques dynamiques du pendule inversé dans des situations plus complexes.

En résumé, la corrélation directe entre le signal sinusoïdal appliqué à l'angle theta et le mouvement sinusoïdal du chariot confirme la validité de la modélisation du système, ouvrant la voie à des analyses plus approfondies et à l'optimisation du contrôle pour maintenir l'équilibre du pendule.

✓ Importation des pièces du système dans l'éditeur de la mécanique en 3D :

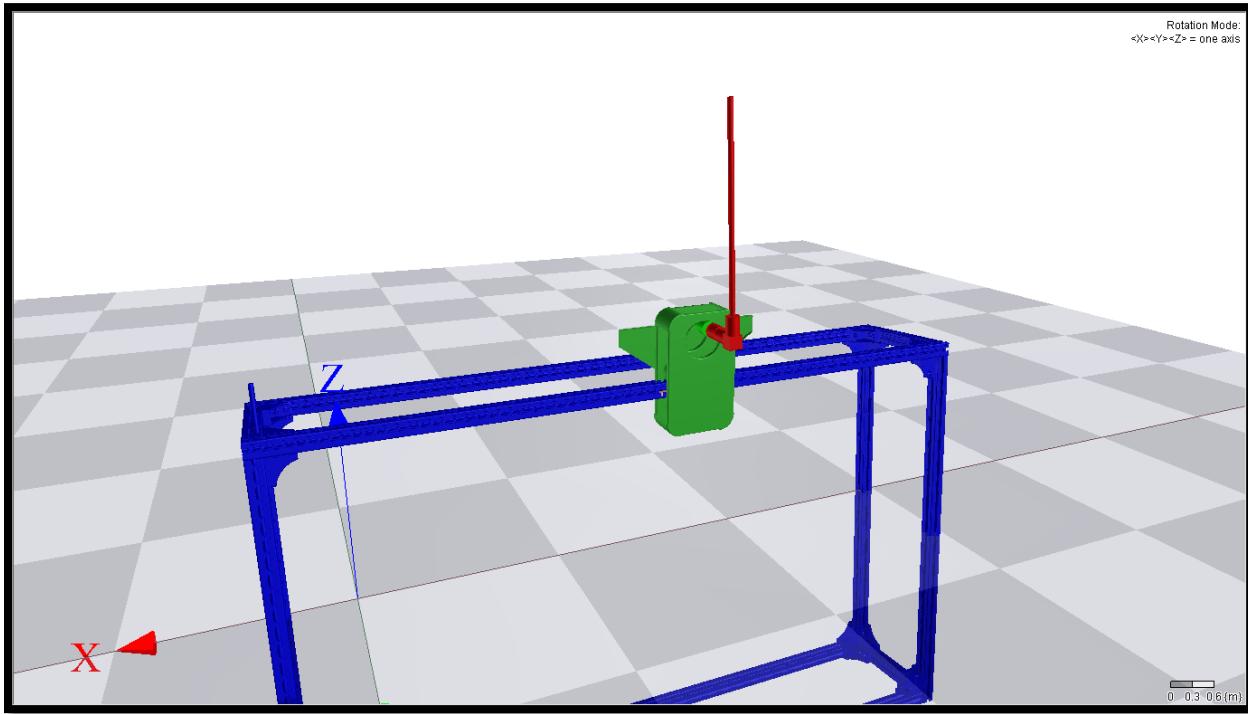


Figure 45: Forme finale du système dans l'éditeur 3D

✓ Génération du code 20-Sim :

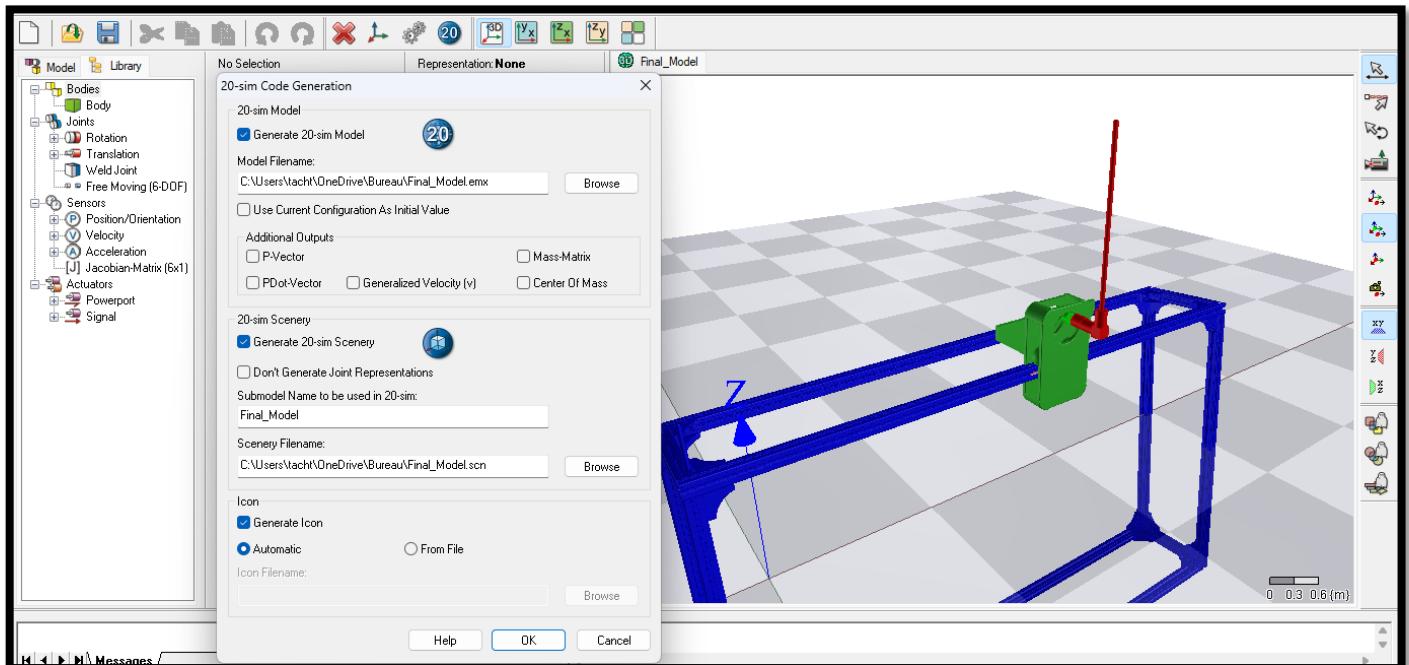


Figure 46: Génération du Code 20-Sim

✓ Model final :

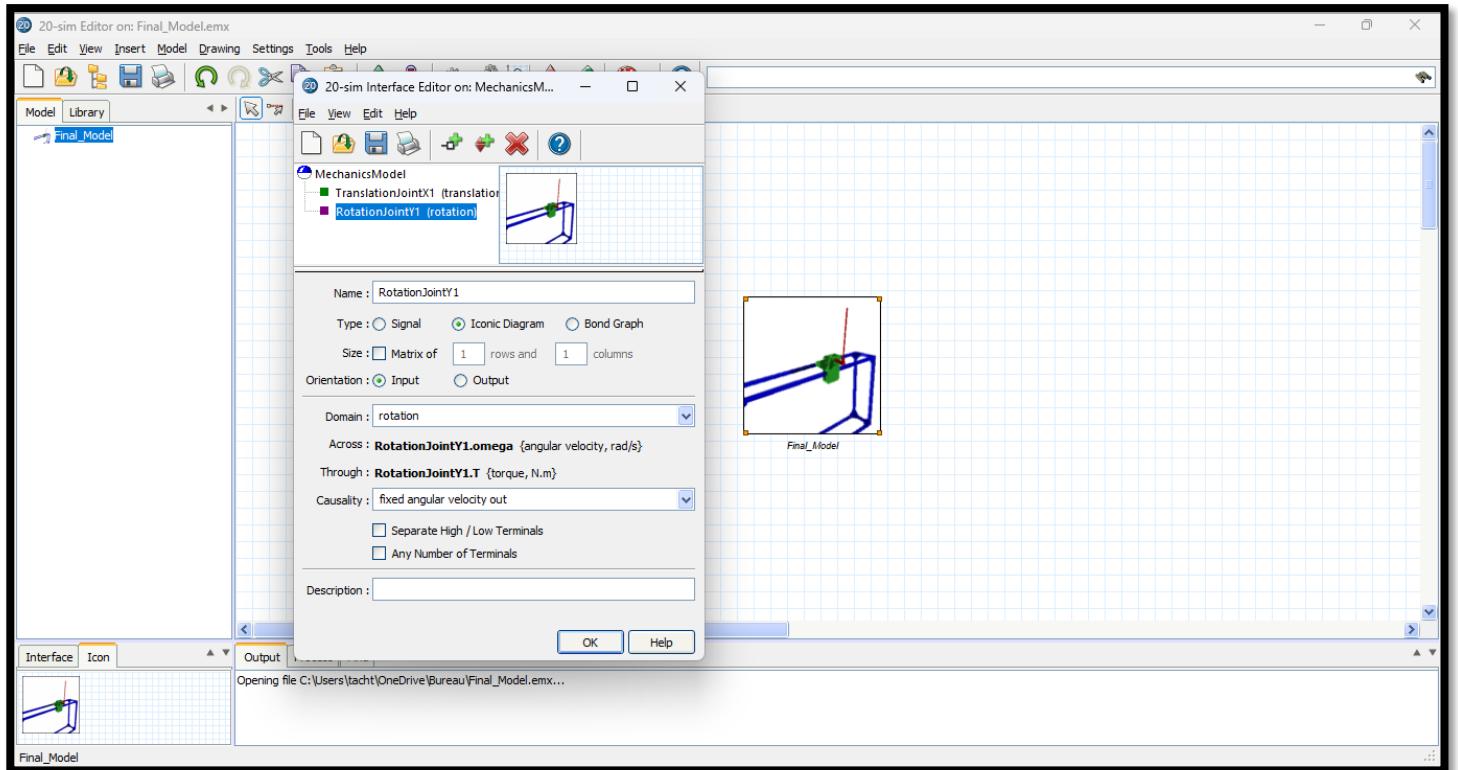


Figure 47: Model Final

Modélisation avec langage python

Après plusieurs discussions avec les membres d'équipe nous avons décidé de faire la modélisation par python

Voici le code que nous avons utilisé :

```

import matplotlib.pyplot as pp
import scipy.integrate as integrate
import matplotlib.animation as animation
from matplotlib.patches import Rectangle

from math import pi
from NumPy import sin, cos, sign

# physical constants
g = 9.8
L = 1.0
m = 0.5

# simulation time
dt = 0.05
Tmax = 35
t = np.arange(0.0, Tmax, dt)

# initial conditions
Y = .0 # pendulum angular velocity
th = pi - 0.1 # pendulum angle
x = .0 # cart position
x0 = 0 # desired cart position
Z = -0.05 # cart velocity
k = 0.08 # control gain coefficient

# Controller coefficients
Kp_th = 50
Kd_th = 15
Kp_x = 3.1
Kd_x = 4.8

state = np.array([th, Y, x, Z])
stabilizing = False

def energy(th, dth):
    return m * dth * L * dth * L / 2 + m * g * L * (cos(th) - 1)

def isControllable(th, dth):
    return th < pi / 9 and abs(energy(th, dth)) < 0.5

def derivatives(state, t):
    global stabilizing
    ds = np.zeros_like(state)

    _th = state[0]
    _Y = state[1] # th'
    _x = state[2]
    _Z = state[3] # x'

    if stabilizing or isControllable(_th, _Y):
        stabilizing = True
    else:
        stabilizing = False

```

```

u = Kp_th * _th + Kd_th * _Y + Kp_x * (_x - x0) + Kd_x * _Z
else:
    E = energy(_th, _Y)
    u = k * E * _Y * cos(_th)

ds[0] = state[1]
ds[1] = (g * sin(_th) - u * cos(_th)) / L
ds[2] = state[3]
ds[3] = u

return ds

print("Integrating...")
# integrate your ODE using scipy.integrate.
solution = integrate.odeint(derivatives, state, t)
print("Done")

ths = solution[:, 0]
Ys = solution[:, 1]
xs = solution[:, 2]
vs = solution[:, 3]

pxs = L * sin(ths) + xs
pys = L * cos(ths)

fig = pp.figure()
ax = fig.add_subplot(111, autoscale_on=False, xlim=(-1.5, 1.5), ylim=(-1.2, 1.2))
ax.set_aspect('equal')
ax.grid()

patch = ax.add_patch(Rectangle((0, 0), 0, 0, linewidth=1, edgecolor='k', facecolor='g'))

line, = ax.plot([], [], 'o-', lw=2)
time_template = 'time = %.1fs'
time_text = ax.text(0.05, 0.9, '', transform=ax.transAxes)

energy_template = 'E = %.3f J'
energy_text = ax.text(0.05, 0.85, '', transform=ax.transAxes)

cart_width = 0.3
cart_height = 0.2

def init():
    line.set_data([], [])
    time_text.set_text('')
    energy_text.set_text('')

    patch.set_xy((-cart_width / 2, -cart_height / 2))
    patch.set_width(cart_width)
    patch.set_height(cart_height)
    return line, time_text, energy_text, patch

def animate(i):
    thisx = [xs[i], pxs[i]]
    thisy = [0, pys[i]]

    line.set_data(thisx, thisy)
    time_text.set_text(time_template % (i * dt))

    E = energy(ths[i], Ys[i])
    energy_text.set_text(energy_template % (E))

    patch.set_x(xs[i] - cart_width / 2)
    return line, time_text, energy_text, patch

```

```
ani = animation.FuncAnimation(fig, animate, np.arange(1, len(solution)),
                             interval=25, blit=True, init_func=in
pp.show
```

D'après l'excusions de ce code nous avons obtenu ce résultat :

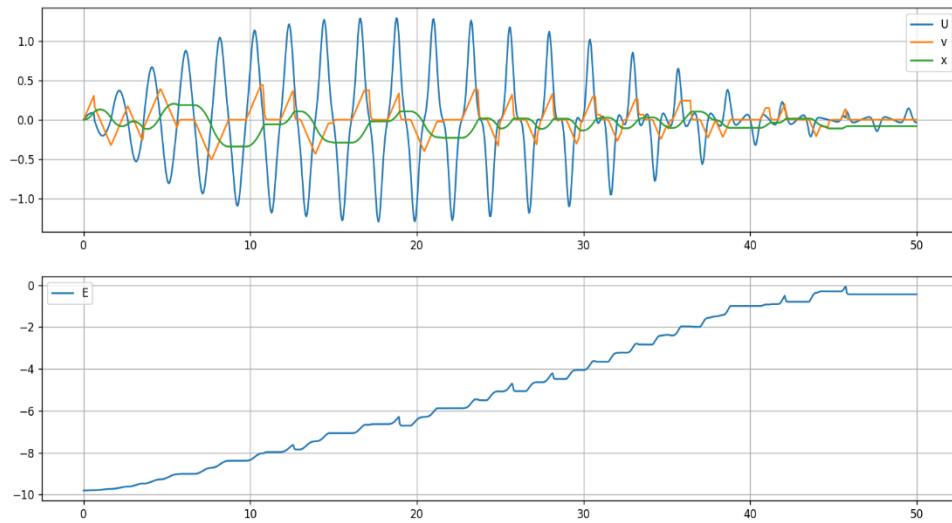


Figure 49: Résultat obtenu

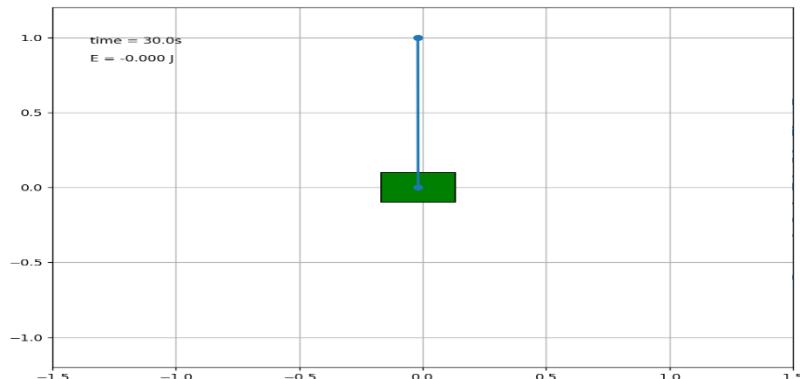


Figure 48: Simulation réelle par Python

CHAPITRE III

Réalisation et tests

I. Matériel utilisé

1. Support

Pour la mise en place de la maquette nous avons utilisé une structure métallique dont la représentation sous SolidWorks est donnée par la figure ci-dessous

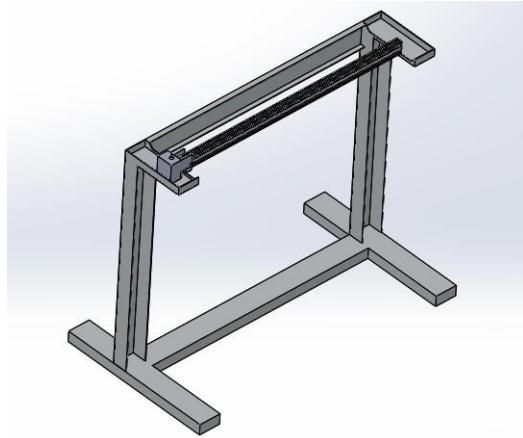


Figure 51: Structure

2. Actionneur

Comme actionneur nous avons utilisé un moteur DC pour assurer le déplacement du chariot à travers un système Poulie-courroie.



Figure 52: Moteur DC

3. Microcontrôleur

Comme microcontrôleur on a utilisé une carte Arduino uno pour l'implémentation du système de contrôle (PID dans notre cas).

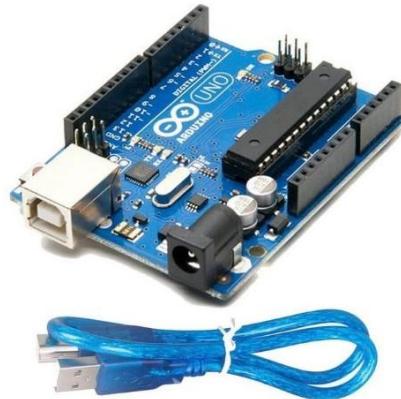


Figure 53: Carte Arduino uno

4. Capteur

Pour mesurer la position angulaire du pendule nous avons opter pour un encodeur rotatif de type LPD3806 600MB.



Figure 54: Encodeur rotatif

5. Chariot

Un chariot constitué de plusieurs pièces en plexiglas, de galets, d'une glissière et d'un pendule constitué par une tige cylindrique en aluminium.

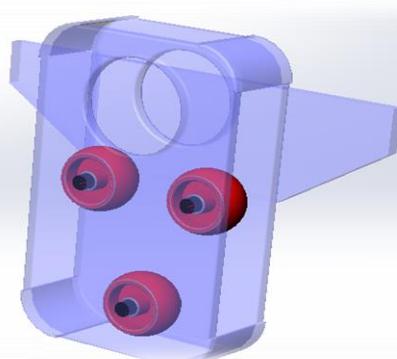


Figure 55 : Chariot

II. Environnement de développement

1) IDE Arduino

L'IDE Arduino (Environnement de développement intégré) est une application logicielle utilisée pour écrire, compiler et télécharger du code sur des microcontrôleurs Arduino. Il offre une interface conviviale pour la programmation et l'interaction avec les cartes Arduino.

On a utilisé l'IDE Arduino pour écrire et implémenter le code de contrôle et de commande sur la carte Arduino.

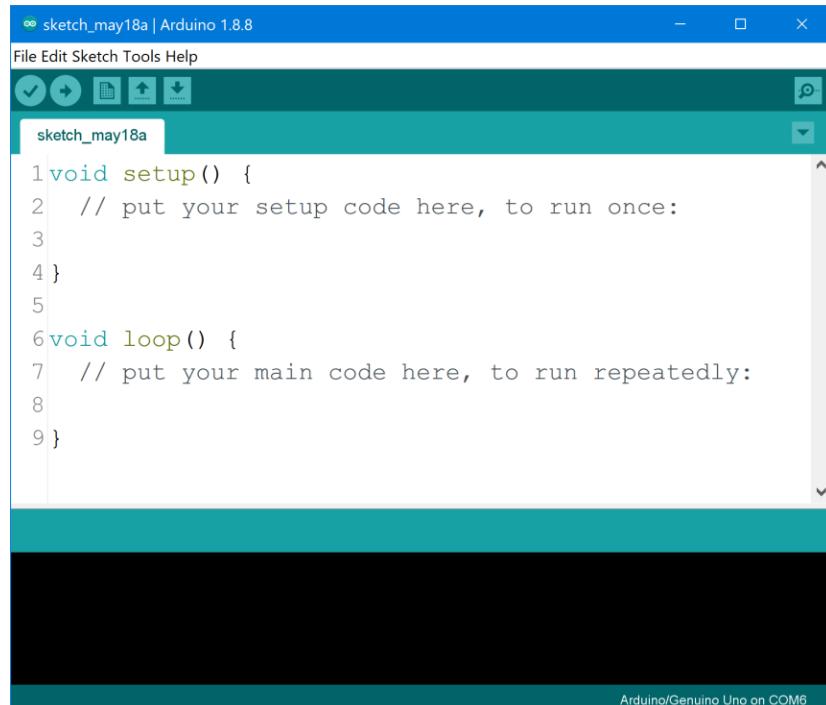


Figure 56: IDE Arduino

2) LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un environnement de développement graphique et un langage de programmation utilisé principalement pour l'automatisation, la mesure et le contrôle des systèmes et des dispositifs. Conçu par National Instruments, LabVIEW permet de créer des applications en combinant des blocs fonctionnels graphiques.

✓ Introduction au Bloc Diagramme LabVIEW pour le Pendule Inverse :

Un bloc diagramme LabVIEW pour le pendule inverse est une représentation graphique de l'algorithme de contrôle mis en œuvre dans LabVIEW pour stabiliser le système du pendule inverse. LabVIEW utilise une approche de programmation graphique, où les fonctions et les opérations sont représentées par des blocs que l'utilisateur peut connecter pour créer un flux de données visuel.

L'objectif du contrôle d'un pendule inverse avec LabVIEW est de stabiliser le système et de maintenir le pendule dans une position d'équilibre malgré les perturbations externes. Plus spécifiquement, les objectifs du contrôle d'un pendule inverse sur LabVIEW incluent :

✓ **Stabilisation du Système :**

L'objectif principal est de concevoir un algorithme de contrôle, généralement basé sur le contrôleur PID, qui stabilise le pendule inverse et le maintient en équilibre autour de la position verticale.

✓ **Optimisation des Paramètres PID :**

Trouver les paramètres PID optimaux pour le contrôleur afin d'assurer une performance maximale du système. Cela peut nécessiter des essais et des ajustements itératifs en utilisant LabVIEW pour optimiser les performances.

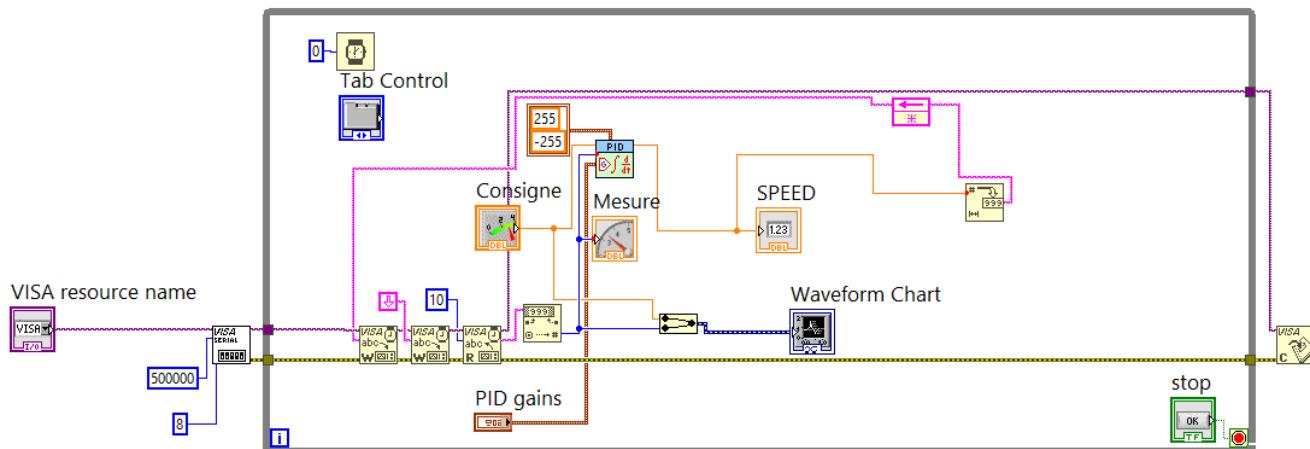
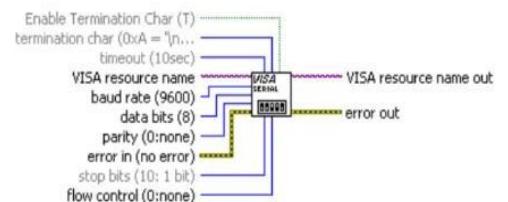


Figure 57: Le bloc Diagramme du code LabVIEW

Dans ce code on a utilisé les Blocs suivants :

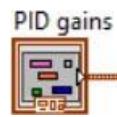
- 1. VISA Configure Serial Port:** ce bloc est utilisé pour configurer les paramètres d'un port série spécifié, tels que la vitesse de transmission, les bits de données, le contrôle de flux, etc., afin de permettre une communication précise et fiable avec des périphériques série.



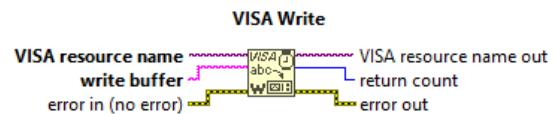
- 2. Consigne :** c'est un contrôleur numérique qui va permettre à l'utilisateur de choisir la position à atteindre.



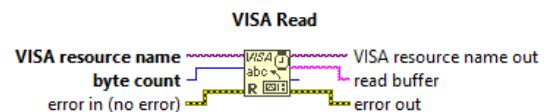
- 3. PID gains :** Le bloc "PID Gains" dans LabVIEW est utilisé pour régler les coefficients de gain du contrôleur PID (Proportionnel, Intégral, Dérivé). Ces paramètres permettent de réguler la performance et la stabilité d'un système decontrôle en ajustant la réponse proportionnelle, intégrale et dérivée du contrôleur.



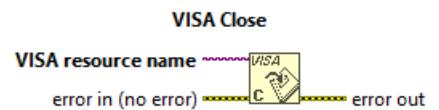
- 4. Visa Write :** Le bloc "VISA Write" dans LabVIEW est utilisé pour envoyer des données sur un port de communication spécifié, tel qu'un port série ou USB, en utilisant la bibliothèque VISA. Il permet d'envoyer des commandes ou des données à des appareils externes de manière simple et efficace.



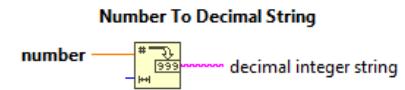
- 5. Visa Read :** Le bloc "VISA Read" dans LabVIEW est utilisé pour lire les données provenant d'un port de communication spécifié, tel qu'un port série ou USB, en utilisant la bibliothèque VISA. Il permet de recevoir et de traiter les réponses ou les données en provenance d'appareils externes de manière efficace et fiable.



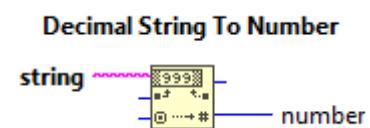
- 6. Visa close :** Le bloc "VISA Close" dans LabVIEW est utilisé pour fermer une session de communication avec un périphérique spécifié à l'aide de la bibliothèque VISA. Il permet de libérer les ressources utilisées par la communication série, USB ou toute autre interface , garantissant une fermeture appropriée de la connexion avec le périphérique.



7. Number to Decimal String : Le bloc "Number To Decimal String" dans LabVIEW est utilisé pour convertir un nombre en une chaîne de caractères représentant sa valeur décimale. Il permet de formater et d'afficher des valeurs numériques de manière lisible, facilitant ainsi la présentation et la manipulation des données numériques dans une interface utilisateur ou lors de la communication avec d'autres composants.



8. Decimal String To Number : Le bloc "Decimal String To Number" dans LabVIEW est utilisé pour convertir une chaîne de caractères représentant une valeur décimale en un nombre numérique. Il permet de récupérer et de traiter des données numériques saisies sous forme de chaîne de caractères, on va l'utiliser pour envoyer les valeurs de l'angle donné par l'encodeur.



9. Waveform chart : Ce Bloc est utilisé pour visualiser pour visualiser la variation de l'angle avec la consigne au même temps en utilisant un multiplexeur.



:

Introduction au Front Panel LabVIEW pour le Pendule Inverse :

Dans LabVIEW, le Front Panel est la partie de l'interface utilisateur où vous créez l'interface graphique de votre application ou de votre programme. C'est l'endroit où vous placez des contrôles (éléments interactifs tels que boutons, cadrants, interrupteurs, etc.) et des indicateurs (éléments affichant des données telles que graphiques, jauge, numériques, etc.) pour interagir avec votre code.

Le Front Panel sert d'interface entre l'utilisateur et le code que vous avez écrit dans le Diagramme LabVIEW. Il vous permet de créer des applications avec une interface utilisateur visuelle et interactive sans avoir à écrire de code graphique.

En résumé, le Front Panel dans LabVIEW est la partie de l'environnement de développement où vous concevez l'interface utilisateur de votre application, facilitant ainsi l'interaction avec votre programme.

Interface Homme-Machine (IHM) :

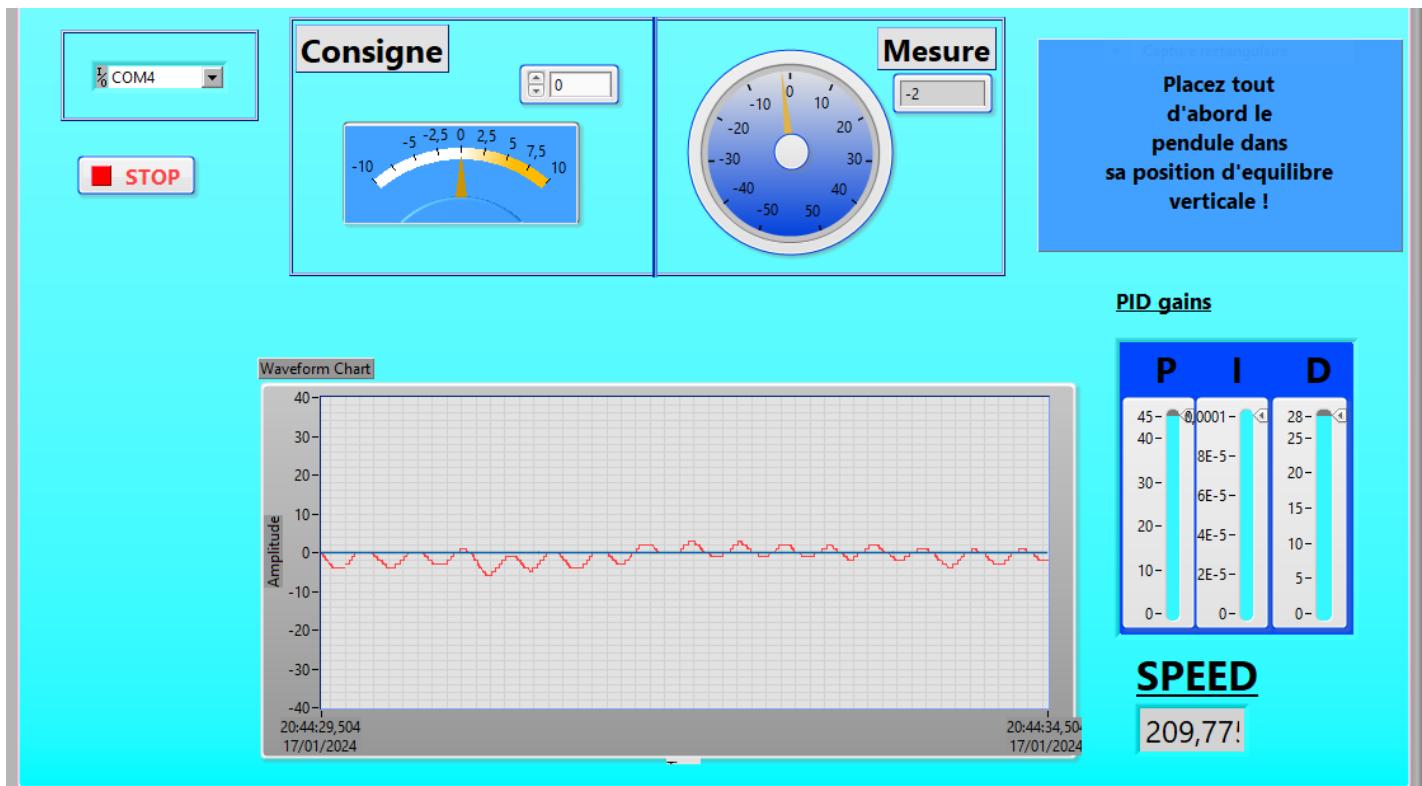


Figure 58: Face avant LabVIEW

➤ Affichage de l'Angle du Pendule

➤ Entrées de Consigne

➤ Paramètres du Contrôleur PID

➤ Boutons de Contrôle

➤ Graphiques de Performance

➤ Contrôle Manuel

L'objectif est de concevoir un panneau avant intuitif qui offre un contrôle complet sur le système, une visualisation claire des informations essentielles et des outils d'analyse pour évaluer les performances du contrôleur PID. LabVIEW offre une flexibilité considérable pour personnaliser l'interface utilisateur en fonction des besoins spécifiques du système de pendule inverse que vous contrôlez.

CONCLUSION

En conclusion, l'utilisation de LabVIEW avec un régulateur PID pour le contrôle du pendule inverse représente une approche puissante et polyvalente. Cette combinaison offre une solution efficace, flexible et évolutive pour le contrôle de systèmes dynamiques, tout en fournissant des outils avancés d'analyse, de visualisation et d'interaction avec le matériel. Elle joue également un rôle important dans l'éducation et la formation dans le domaine du contrôle automatique.

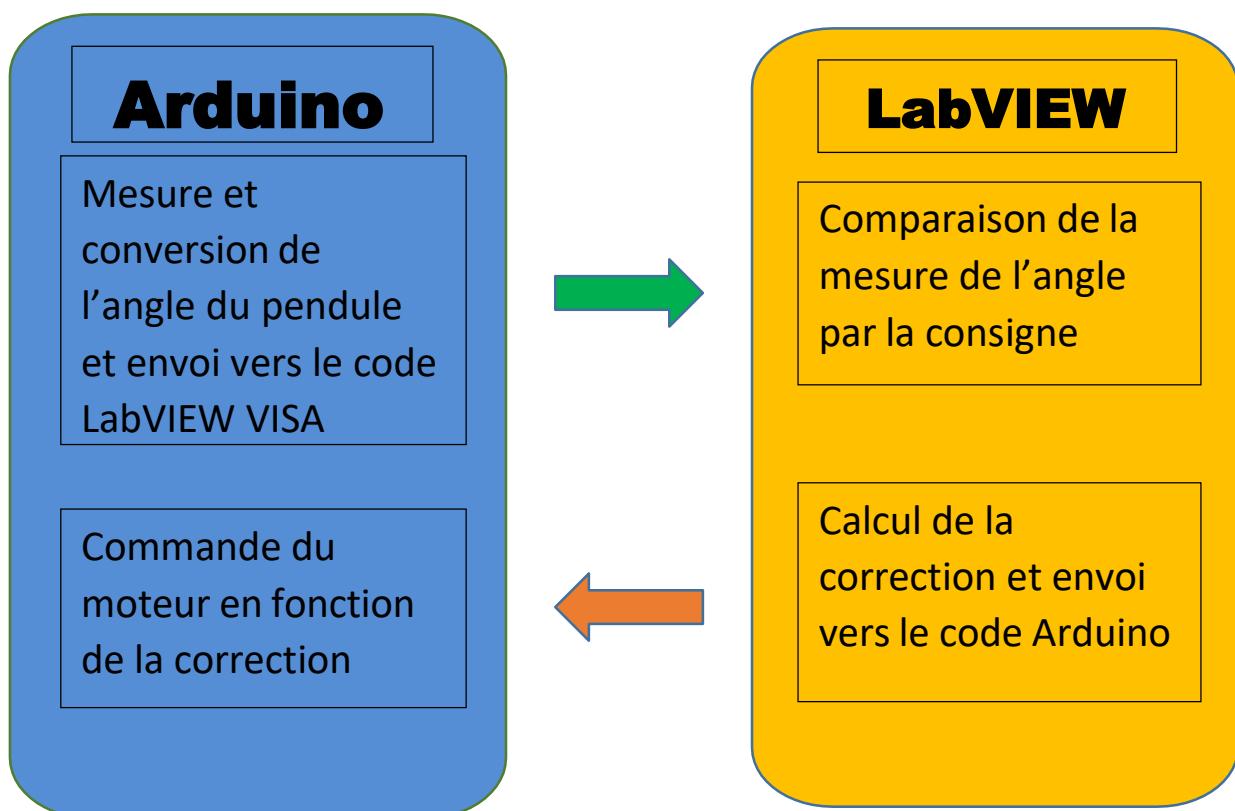


Figure 59: illustration de la Communication Arduino- LabVIEW

Conclusion Générale

En conclusion, le pendule inversé représente un modèle fascinant dans le domaine du contrôle automatique et de la dynamique des systèmes. Sa stabilité instable offre des défis stimulants pour les ingénieurs et les chercheurs, poussant le développement de stratégies de contrôle avancées. Les applications pratiques de ces connaissances s'étendent à des domaines tels que la robotique, la stabilisation de systèmes autonomes et la formation pédagogique. L'étude du pendule inversé continue d'être un terrain fertile pour la recherche, ouvrant la voie à des innovations significatives dans la conception et la mise en œuvre de systèmes de contrôle dynamique. Cette exploration approfondie du pendule inversé contribue ainsi à l'évolution constante des technologies de contrôle et à la résolution de problèmes complexes dans divers secteurs professionnels.